

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-359850

(P2002-359850A)

(43) 公開日 平成14年12月13日 (2002. 12. 13)

(51) Int.Cl. ⁷	識別記号	F I	データコード (参考)
H 0 4 N 7/30		G 0 6 F 17/14	S 5 B 0 5 6
G 0 6 F 17/14		H 0 3 M 7/30	A 5 C 0 5 9
H 0 3 M 7/30		7/40	5 C 0 7 8
7/40		H 0 4 N 1/41	B 5 J 0 6 4
H 0 4 N 1/41		7/133	Z
審査請求 未請求 請求項の数22 O L (全 30 頁)			

(21) 出願番号 特願2002-35886(P2002-35886)

(22) 出願日 平成14年2月13日 (2002. 2. 13)

(31) 優先権主張番号 7 8 4 9 2 8

(32) 優先日 平成13年2月15日 (2001. 2. 15)

(33) 優先権主張国 米国 (U S)

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者 エドワード エル シュワルツ

アメリカ合衆国、カリフォルニア州

94025, メンロ・パーク, サンド・ヒル・

ロード 2882番, スイート 115, リコー

イノベーション内

(74) 代理人 100070150

弁理士 伊東 忠彦

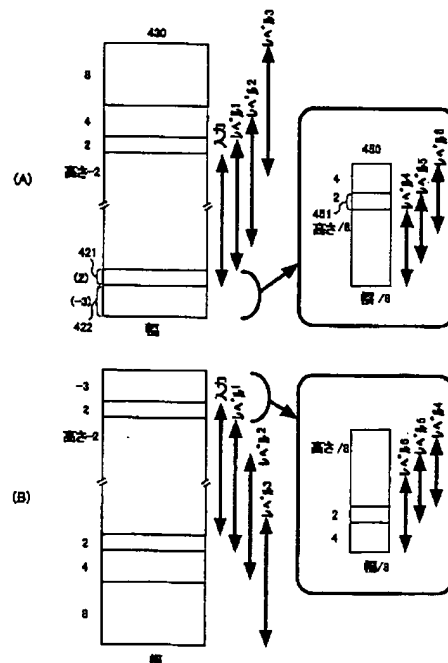
最終頁に続く

(54) 【発明の名称】 ウェーブレット処理を実行するためのメモリ使用機構

(57) 【要約】

【課題】 本発明は、高速、低コスト、小メモリで特徴の豊富な実行を可能とする圧縮と逆圧縮機構の改善を行うことを目的とする。

【解決手段】 メモリとウェーブレット処理論理を含むシステムが開示される。メモリは画像のバンドと追加のラインを蓄積するために複数のラインを含むサイズを有する。ウェーブレット処理論理はウェーブレット変換とアクセス論理を含む。ウェーブレット変換はメモリ内のデータに行われたときに係数を発生する。アクセス論理は、メモリに記憶されたデータをウェーブレット変換に供給し且つ、メモリからアクセス論理のバッファへ第1の組のラインに記憶されたデータが読み出された後にメモリ内に係数を蓄積するために、メモリからラインバッファ内にデータを読出す。アクセス論理は、第1の対のラインと異なる第2の対のラインに関連しているウェーブレット変換により発生された係数を記憶するために、第1の対のラインを再使用する。



【特許請求の範囲】

【請求項1】 画像の帯を記憶するラインと追加の複数のラインを含むようなサイズのメモリを有し、メモリ内のデータに適用されたときに係数を発生するウェーブレット変換と、

メモリ内に記憶されたデータをウェーブレット変換に供給するラインバッファへ、メモリからデータを読み出し、且つ、メモリへ、係数を記憶するアクセス論理とを有し、アクセス論理のバッファへ、メモリから、第1の対のラインに記憶されたデータが読み出された後に、アクセス論理は、第1の対のラインと異なる第2の対のラインに関連するウェーブレット変換により発生された係数を記憶する第1の対のラインを再使用する、ウェーブレット処理論理を有するシステム。

【請求項2】 アクセス論理は、メモリの連続するライン内に、係数を、同じサブバンドで且つ互いに隣接する分解レベルからの係数と共に記憶する、請求項1に記載のシステム。

【請求項3】 第1と第2の対のラインの各々の第1のラインは、互いにオフセットしてメモリ内に配置されている請求項1に記載のシステム。

【請求項4】 アクセス論理は、オフセットの距離内の追加のライン内の各係数レベルに対するウェーブレット変換の第1の出力を記憶する請求項3に記載のシステム。

【請求項5】 オフセットのサイズは、各変換レベルで異なる請求項3に記載のシステム。

【請求項6】 オフセットのサイズは、2
記憶された係数の変換レベル
に等しい請求項3に記載のシステム。

【請求項7】 分解中に、メモリ内にL1係数の行の各対の第1の行を記憶するためのオフセットは、L1係数の行の前記各対に関連する画像のデータの第1の行から2ラインであり、L2係数の行の各対の第1の行を記憶するオフセットは、L2係数の行の前記各対に関連するL1係数の第1の行から4ラインである、請求項6に記載のシステム。

【請求項8】 アクセス論理は、前に画像の帯を記憶したメモリのライン内のレベル3よりも大きい分解レベルに関連する係数を記憶する、請求項1に記載のシステム。

【請求項9】 オフセットに関連する追加のラインは、画像の帯を記憶するラインより上である請求項3に記載のシステム。

【請求項10】 ウェーブレット変換は、前向きウェーブレット変換である請求項1に記載のシステム。

【請求項11】 ウェーブレット変換は、逆ウェーブレット変換である請求項1に記載のシステム。

【請求項12】 ウェーブレット変換を適用するためにラインバッファ内にメモリからデータを読み出し、且つ、

メモリ内の各対のラインに記憶されたデータから発生された係数の各組が、メモリ内の前記各対のラインへ要求するオフセットで、メモリ内の各ラインに記憶されるように、メモリ内のラインにウェーブレット変換を適用することにより形成された係数を記憶する、方法。

【請求項13】 第1の対のラインに記憶されたデータがアクセス論理のバッファにメモリから読み込まれた後に、ウェーブレット変換により発生された係数を記憶するために、第1の対のラインと異なる第2の対のラインと関連する、第1の対のラインを再使用するアクセス論理を更に有し、且つ、第1と第2の対のラインの各々の第1のラインは、互いに関してオフセットしてメモリ内に配置される、請求項12に記載の方法。

【請求項14】 オフセットの距離内の追加のライン内の各係数レベルに対するウェーブレット変換の第1の出力を記憶するアクセス論理を更に有する請求項13に記載の方法。

【請求項15】 オフセットのサイズは、各変換レベルに対して異なる請求項13に記載の方法。

【請求項16】 オフセットのサイズは、2
記憶された係数の変換レベル
に等しい請求項13に記載の方法。

【請求項17】 分解中に、メモリ内にL1係数の行の各対の第1の行を記憶するためのオフセットは、L1係数の行の前記各対に関連する画像のデータの第1の行から2ラインであり、L2係数の行の各対の第1の行を記憶するオフセットはL2係数の行の前記各対に関連するL1係数の第1の行から4ラインである、請求項16に記載の方法。

【請求項18】 前に画像の帯を記憶したメモリのライン内のレベル3よりも大きい分解レベルに関連する係数を記憶するアクセス論理を更に有する、請求項12に記載の方法。

【請求項19】 オフセットに関連する追加のラインは、画像の帯を記憶するラインより上である請求項13に記載の方法。

【請求項20】 実行可能な命令を蓄積する少なくとも1つの記録可能な媒体を有する製造物であって、処理装置により実行されたときに、

ウェーブレット変換を適用するラインバッファ内にメモリからデータを読み出し、且つ、メモリ内の各対のラインに記憶されたデータから発生された係数の各組が、メモリ内の前記各対のラインへ要求するオフセットで、メモリ内の各ラインに記憶されるように、メモリ内のラインにウェーブレット変換を適用することにより形成された係数を記憶することを処理装置に行わせる製造物。

【請求項21】 処理装置により実行されたときに、処理装置に、第1の対のラインに記憶されたデータが、アクセス論理のバッファにメモリから読み込まれた後に、

10

20

30

40

50

ウェーブレット変換により発生された係数を記憶するために、第1の対のラインと異なる第2の対のラインと関連する、第1の対のラインを再使用することを行わせ、且つ、第1と第2の対のラインの各々の第1のラインは、互いに関してオフセットしてメモリ内に配置される命令を、更に有する請求項20に記載の製造物。

【請求項22】 ウェーブレット変換を適用するラインバッファ内にメモリからデータを読出す手段、且つ、メモリ内の各対のラインに記憶されたデータから発生された係数の各組が、メモリ内の前記各対のラインへ要求するオフセットで、メモリ内の各ラインに記憶されるように、メモリ内のラインにウェーブレット変換を適用することにより形成された係数を記憶する手段とを有する装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、圧縮と逆圧縮の分野に関連し、特に本発明は、ウェーブレット処理を実行するためのメモリの使用に関連する。

【0002】

【従来の技術】新しいJPEG2000復号規格(ITU-T Rec. T. 800/ISO/IEC15444:2000 JPEG2000画像符号化システム)は、画像に対する新たな符号化機構と符号ストリームの定義を提供する。JPEG2000規格は、復号規格であるが、JPEG2000は復号器が行わねばならないことを定義することにより符号化と復号を規定する。JPEG2000規格の下では、各画像は1つ又はそれ以上の矩形のタイル(領域)に分割される。1つ以上のタイルがある場合には、画像のタイル化は、互いに独立して抽出され又は復号されるタイル成分を生成する。タイル成分は、タイル内の所定の成分の全てのサンプルを有する。画像は複数の成分を有する。そのような成分の各々は、2次元配列のサンプルを有する。例えば、カラー画像は赤、緑及び、青色成分を有する。

【0003】画像のタイル化後に、タイル成分は、ウェーブレット変換を使用して異なる分解レベルに分解される。これらの分解レベルは、元のタイル成分の水平及び垂直空間周波数を記述する係数で占められた幾つかのサブバンドを含む。この係数は、画像全体をわたってではなく、局部領域についての周波数情報を提供する。即ち少数の係数は完全に単一のサンプルを記述する。分解レベルは、サブバンドの各々の連続する分解レベルが、前の分解レベルの約半分の水平解像度と半分の垂直解像度を有するように、2の空間係数により次の分解レベルと関連する。

【0004】サンプルと同数の係数があるけれども、情報内容は幾つかの係数のみに集中する傾向にある。量子化を通して、多くの数の係数は更に減少される。エントロピー符号化器による追加の処理は、これらの量子化さ

れた係数を表すのに必要なビット数を、しばしば元の画像と比較して大きく、減少する。

【0005】タイル成分の個々のサブバンドは、更に、コードブロックに分割される。これらのコードブロックは、パーティションにグループ化される。係数のこれらの矩形配列は、独立に抽出される。コードブロック内の係数の個々のビット面は、3つの符号化パスでエントロピー符号化される。これらの符号化パスの各々は、圧縮された画像データのビット面に関する前後関係上の情報を集める。

【0006】これらの符号化パスから形成された、圧縮された画像データのビットストリームは、階層にグループ化される。階層は、コードブロックからの連続する符号化パスの任意のグループ化である。階層化には大きな柔軟性があるが、前提は、各々の連続する階層がより高い品質の画像に貢献することである。各解像度レベルにおけるサブバンド係数は、区画と呼ばれる矩形領域に分割される。

【0007】パケットは、圧縮されたコードストリームの基本的な単位である。パケットは、1つのタイル成分の1つの解像度レベルの1つの区画の1つの階層からの圧縮された画像データを含む。これらのパケットは、コードストリーム内で定義された順序で配置される。

【0008】パケットに組織化されたタイルに関連するコードストリームは、1つ又はそれ以上のタイル部分に配置される。一連のマーカー及びマーカーセグメント又はタグより構成されるタイル部分ヘッダーは、各タイル毎に位置を決め、抽出し、復号しそして再構成するために必要な、種々の機構と符号化形式に関する情報を含む。全コードストリームの先頭には、マーカーとマーカーセグメントよりなり、オリジナル画像に関する情報と同様な情報も提供するメインヘッダーがある。

【0009】コードストリームは、アプリケーションが画像の意味とそれに関する他の情報を翻訳することを可能とする、ファイルフォーマット内にの含まれることができる。ファイルフォーマットは、コードストリームの他のデータを含んでも良い。

【0010】

【発明が解決しようとする課題】JPEG2000コードストリームは、符号化ステップでの順序を逆にすることにより実行される。図1は、圧縮された画像データコードストリームに関して作用する、JPEG2000規格符号化機構のブロック図を示す。図1を参照すると、ビットストリームは最初に、階層とサブバンド係数を再グループ化するデータ整列ブロック101により受信される。算術符号化器102は、圧縮されたビットストリームを復号するために、ビット面圧縮された画像データに関する符号化中に集められた関係情報とその内部状態を使用する。

【0011】算術復号の後に、係数は、係数ビットモデ

ル化ブロック103で、ビットモデル化をうける。次にコードストリームは、ROIブロック105で示されている対象の領域(ROI)に基づいて量子化される、量子化ブロック104で量子化される。量子化の後に、変換ブロック106を介して、残りの係数に逆変換が行われ、続いて、DC及びオプションの成分変換ブロック107が続く。これで、再構成された画像が発生される。

【0012】JPEG2000規格は実行のために多くの選択が残っている。

【0013】

【課題を解決するための手段】メモリとウェーブレット処理理論を含むシステムが開示される。メモリは、画像の帯(バンド)と追加のラインを蓄積するために複数のラインを含むサイズを有する。ウェーブレット処理理論は、ウェーブレット変換とアクセス論理を含む。ウェーブレット変換は、メモリ内のデータに行われたときに係数を発生する。アクセス論理は、メモリに記憶されたデータをウェーブレット変換に供給しそして、メモリからアクセス論理のバッファへ第1の対のラインに記憶されたデータが読み出された後にメモリ内に係数を蓄積するために、メモリからラインバッファ内にデータを読み出す。アクセス論理は、第1の対のラインと異なる第2の対のラインに関連しているウェーブレット変換により発生された係数を記憶するために、第1の対のラインを再使用する。

【0014】

【発明の実施の形態】本発明の添付の図面と以下の詳細な説明から完全に理解されよう。しかしながら本発明を、特定の実施例は、説明と理解のためのみであり本発明を制限するものと解してはならない。

【0015】圧縮と逆圧縮機構の改善が開示される。高速、低コスト、小メモリ及び／又は、特徴の豊富な実行を行うために、JPEG2000内の選択項目を使用することが、ここに開示される技術と実行の目的である。

【0016】以下の説明では、本発明の完全な説明を行うために、多くの詳細が述べられる。しかし、当業者には、本発明が、これらの特定の詳細無しに実行できることは明らかである。例えば、既知の構造と装置は、本発明を曖昧にすることを避けるために、細部の代わりに、ブロック図の形式で示される。

【0017】以下の詳細な説明の幾つかの部分は、コンピュータメモリ内のデータビットに関する動作のアルゴリズム及び記号表現で提示される。これらのアルゴリズム的な記載と表現は、他の当業者に研究の実体を最も効率的に伝えるための、データ処理技術の分野の当業者により使用される手段である。アルゴリズムは、ここでは、所望の結果を導く自己一貫性のあるステップのシーケンスであると考えられる。これらのステップは、物理的な量の物理的な操作を必要とする。通常は、不用であるが、これらの量は、蓄積され、伝送され、結合され、

比較されそして他の操作が行われることが可能な、電気的又は磁気的な信号の形式を取る。主に共通の使用の理由から、これらの信号を、ビット、値、要素、シンボル、キャラクター、項、数又は、同様なものとして参照するのが、しばしば便利であることが分かる。

【0018】しかし、すべてのこれらの同様な用語は、適切な物理量に関連しており、そして、これらの量に付される単なる便利なラベルであるということは記憶すべきである。以下の説明から明らかなように特に述べない限り、“処理する”又は“計算する”又は、“算出する”又は、“決定する”又は、“表示する”又は、同様なもののような説明を通しての用語の使用は、コンピュータシステムレジスタ及びメモリ内の物理的な(電気的な)量として示されるデータを、コンピュータシステム及びメモリ又はレジスタ又は他の情報記憶、伝送又は、表示装置内の物理的な量として同様に示される他のデータに操作し且つ変換する、コンピュータシステム又は同様な電子計算器の動作及び処理に参照されることは理解される。

【0019】本発明は、ここで説明する動作を実行する装置にも関連する。この装置は、必要な目的のために特別に構成され又は、コンピュータ内に蓄積されたコンピュータプログラムにより選択的に活性化され又は再構成される汎用コンピュータを含んでも良い。そのようなコンピュータプログラムは、そして各々はコンピュータシステムバスに接続された、制限はされないが、フレキシブルディスク、光ディスク、CD-ROM及び、磁気ディスク、読出し専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、EPROM、EEPROM、磁気又は光カード又は、電子的命令を記憶するのに適する他の形式の媒体のような、コンピュータ読出し可能な蓄積媒体に記憶される。

【0020】ここで示すアルゴリズムと表示は、特定のコンピュータ又は、他の装置に固有に関連してはいない。種々の汎用システムが、ここで説明する技術に従ってプログラムと共に使用でき、又は、要求される方法ステップを実行する更に特化された装置を便利に実現し得る。これらの種々のシステムに対する必要な構造は、以下の説明から明らかとなろう。更に加えて、本発明は、特定のプログラム言語を参照して記載されていない。種々のプログラム言語は、ここで開示する本発明の技術を実行するのに使用できることは理解されよう。

【0021】機械で読出し可能な媒体は、機械(例えば、コンピュータ)により読出し可能な形式の、情報を記憶し且つ伝送する任意の機構をふくむ。例えば、機械で読出し可能な媒体は、読出し専用メモリ(“ROM”)、ランダムアクセスメモリ(“RAM”)、磁気ディスク蓄積媒体、光蓄積媒体、フラッシュメモリ装置、電気的、光学的、音響的又は、他の形式の伝播信号(例えば、搬送波、赤外信号、デジタル信号等)等を

含む。

【0022】概観

以下の説明は、JPEG2000又は、同様な特徴の他の符号化機構の柔軟性を利用する実行又は新しい方法に関連する。

【0023】低メモリ及び高速バーストアクセスに関するメモリ使用

図2は、メモリ201内の画像の構成の一実施例を示す。図2を参照すると、“タイル高さ”ラスタラインのみ又は、画像の帯（バンド）がメモリ201内にあり、全体画像はない。このように、メモリ201内の画像の量は、画像幅掛ける画像高さに等しい。画像のバンドの内側には、タイル210のような少なくとも1つのタイルがある。

【0024】ウェーブレット変換処理論理202は、データがウェーブレット変換202Bに与えられるように、メモリ201からデータを読みそしてメモリ201にデータを記憶するメモリアクセス論理202Aを有する。ウェーブレット変換処理論理202は、ハードウェア、ソフトウェア又は、その両者の組み合わせを有する。

【0025】一実施例では、メモリアクセス論理202Aは、タイルに、メモリ内のタイルの先頭へのポインタ又はインデックス、タイルの幅、タイルの高さ及び、1つのラインの先頭から他のラインの先頭を得るためのラインオフセット（画像幅）の、4つのパラメータでアクセスする。代わりに、メモリアクセス論理202Aは、タイルの幅の代わりにタイルの最後へのポインタ又はインデックスを使用してメモリ201にアクセスできる。

【0026】一実施例では、ある関数Fを実行するために、タイルの各ライン又は、画像のラインの一部にアクセスするために、以下の処理が使用される。

【0027】

【数1】

```
line = start
for y = 0 to tile_height - 1
  for x = 0 to tile_width - 1
    perform function F with line[x]
  line = line + line_offset
```

関数の1つFは、ラインの組みにウェーブレット変換を与えることを含む。他の関数Fは、DCレベルシフト、多重成分変換でも良い。

【0028】そのような処理は、ハードウェア（例えば、専用の論理、回路等）、（汎用コンピュータ又は、専用機械で実行される）ソフトウェア、又は、その両者の組合せを含む処理論理により行われる。

【0029】一実施例では、サブバンドからの係数は、開始点、幅、高さ及び、ラインオフセットを有する同様な機構を使用してアクセスされる。係数の行はともにメモリ内に蓄積されるので、キャッシュ、バーストアクセ

ス可能メモリ又は、1係数よりも広いメモリ内に記憶されたときには効率的にアクセスされる。

【0030】図3（A）－（C）は、係数がどのように記憶されるかを概念的に示す、種々のレベルに対する変換メモリの構成を示す。（ITU-T勧告800/ISO/IEC15444-1:2000JPEG2000画像符号化システムの命名法を使用する）全てのLH、HL及び、HH係数は、符号化される。これらの係数は、図3（B）と3（C）内では破線で示される。符号化されている最終の変換レベルのLL係数を除いて、変換を計算しているときには、入力タイル301の入力ラインと（連続するレベルの図3（B）と3（C）内で実線で示された）LL係数のみが、一時的に記憶される必要がある。水平と垂直を1つのパスで行い且つラインバッファを使用する変換が使用されているときには、一旦、入力行の組みが完全に読み込まれ（入力行又は、LL係数）、ラインにより使用される空間は再使用される。

【0031】図3（A）－（C）は、一実施例内でどのように際しようが達成されるかを示すために、入力タイル301、オフセットで整列されたレベル1（L1）（302）及びレベル2（L2）（303）メモリ領域を示す。入力タイル301を保持するのに使用されるメモリ空間への行312と313の2つの行の付加は、L1係数に対する入力タイル301に対するメモリを再使用するとき、L1係数を発生するのに必要とされる。L1係数を保持するのに使用されるメモリ空間への、行341－342の4つの行の付加は、L2係数に対するL1係数を記憶するメモリメモリを再使用するとき、L2係数を発生するのに必要とされる。（行341と342の間に、無駄な空間の2つの行があることに注意する。）追加のラインは、ウェーブレット変換がメモリ内の情報に与えられている方向の後ろにあるのが好ましい。

【0032】特に、入力タイル301の入力行の組みは、2つの異なるラインへの変換を与えた結果のそして、メモリのラインに書きこまれているウェーブレット変換を与えた結果の、レベル1のLL、LH、HL及びHH係数の各々の1つの行に対応する。例えば、入力行310と311へウェーブレット変換を与えた結果は、L1係数（302）の行312と313の部分内の係数である。例えば、行312のLL係数321はレベル1のLL係数（実線）に対応し、行312のHL係数322はレベル1のHL係数に対応し、行313のLH係数323はレベル1のLH係数に対応し、そしてHH部分324はレベル1のHH係数に対応する。最初の2つの入力ラインからのレベル1係数はメモリの一番上の2つの余分な行内に記憶され、残りのレベル1係数はメモリを再使用するために入力タイル301のデータを記憶する位置に書きこまれる。レベル1係数の各サブバンドに対する係数の各形式（例えば、LH、HL、HH）に

10

20

30

40

50

対する幅と高さは、入力タイル301の半分である。レベル1に関するLL行から次のLL行のラインオフセット（例えば、図3（B）内の行312から行314の距離）は、（各係数行は2つのラインに対応する領域からであるので）タイル幅の2倍である。

【0033】同様に、レベル1でのLL係数の2つの行にウェーブレット変換を与えた結果（実線）は、2つの行内の係数である、即ち、レベル2での、LL係数（331）、LH係数（332）、HL係数（333）及び、HH係数（334）である。レベル2係数の幅と高さは、入力タイル301の4分の1である。レベル2に対するラインオフセットは、（各係数行が、2つのレベル1LL行又は、4つの入力ラインに対応する領域からであるので）タイル幅の4倍である。このように、メモリの4つの余分なラインが、L2係数を記憶するために入力タイルを記憶する同じメモリを使用するのに必要とされる。第3の分解レベルが実行された場合には、追加の8ラインが必要なことに注意する。従って、この例では、入力時間を記憶し且つそこに与えられる分解の2つのレベルを有するメモリを再使用することを可能とする

ために、合計で14の余分なラインが必要とされる。余分なラインを決定するのに子よされる一般的な公式は次の、

$$2^{(\max \text{level} + 1)} - 2$$

で与えられる。

【0034】LL、LH、HL及び、HHサブバンドのような、サブバンドにアクセスするために、先頭ポイントと行/ライン間のオフセットが必要である。高さとも、タイルをアクセスするときに停止するときを知るために必要である。

【0035】分解レベルが増加すると、メモリ内の幾つかの行が未使用となる。即ち、第1の分解レベル後のL1係数の下のメモリのラインが未使用となり、第2の分解レベル後のL2係数の下のメモリのラインが未使用となる等である。一実施例では、余分な空間が再使用される。

【0036】図3（D）-3（F）は、追加のラインが逆変換を与えた結果を記憶する、そして、これらの追加のラインはメモリ内で逆変換が行われている方向の後ろにある、対応する逆変換メモリ使用を示す。

【0037】図4（A）は、入力と種々の分解レベルをそれぞれ前向き変換に記憶できる単一メモリの実施例を示す図である。図4（A）を参照すると、入力タイル、レベル1係数、レベル2係数及び、レベル3係数に対する位置が、それぞれ、追加の2、4及び、8ラインに示されている。図4（B）は、変換の種々のレベルの入力係数が逆変換中に出力と共に記憶できる、同様な単一メモリの実施例を示す図である。

【0038】表1は、別々のメモリと再使用されたメモリに対する、256x256タイルの種々の変換レベル

に対して要求されるメモリの量を示す。

【0039】

【表1】

表1

レベル	別のメモリ (バイト)	再使用されたメモリ (バイト)
1	256x256 = 65,536	2x256 = 512
2	128x128 = 16,384	4x256 = 1,024
3	64x64 = 4,096	8x256 = 2,048
4	32x32 = 1,024	16x256 = 4,096
5	16x16 = 256	32x256 = 8,192
6	8x8 = 64	64x256 = 16,384

再使用されたメモリに対して、リストされた量はそのレベルに対して使用される追加の新たなメモリである。この例に対して、レベル1、2及び、3に対するメモリの再使用はメモリを節約する。レベル4は別のメモリを使用する。

【0040】レベル4、5及び、6に対するメモリは、レベル3が発生された後の単一メモリ内に又は、完全に異なる且つ別のメモリ内に配置することができる。必要なメモリの量は、38x32であり、それは、5x256よりも小さい。レベル1係数の発生後に2つの未使用ライン（即ち、入力データの最後の2つのラインを記憶したメモリ）があるので、小メモリの節約は、これらの2つのラインをレベル4、5及び、6で再使用させることにより達成できる。レベル4、5及び、6に対する追加のライン数は、16、32及び64であり、且つライン間の余分な空間は、前のレベルより2倍離れており且つ半分の幅であるので、これは、特に重要である。

【0041】一実施例では、レベル4、5及び、6からの係数は、図4の記憶領域450のような小メモリ構造に詰められる。図4を参照すると、レベル4係数は、8（3レベルでは 2^3 ）で割ったタイル高さに等しい高さを有し、8（ 2^3 で、3は前に記憶されたレベルの数に対応する）で割った幅wと等しい幅を有する領域内に記憶される。追加の2つのライン451は、同じ必要な記憶領域内にレベル5の係数を記憶するのに、全て必要である。同様に、追加の4ラインは、レベル6の係数に対してこのメモリ記憶領域を使用して収容するのに、全て必要である。係数を記憶するときに、ラインをスキップしないことに注意する。256x256タイルが処理されている、一実施例では、蓄積領域430の低部に余分な5ライン、2ライン421と約4.75ライン422は、記憶領域450を収容するのに使用される。示されたように、3ライン422による近似は、割当てられたメモリ又は、それに加えて、入力タイルを記憶するのに必要なメモリを示す。この方法で、入力タイルを記憶する領域はほぼ完全に再使用される。

【0042】一実施例では、非常に少しの又は潜在的に最小のメモリを使用するために、レベル6は、レベル4と5から別に記憶される。しかし、これは64バイトの

メモリを節約するだけである。

【0043】273×256よりも少し小さなメモリは、256×256タイルの全ての変換係数を保持できる。これは、真に適切なメモリ構成よりも7%小さい。適切なメモリ構成と違って、余分なコピーが避けられ一方、同時に、高速アクセスのために共に詰められた行を保持する。

【0044】表2は、128×128タイルに対する、別のメモリと再使用されたメモリの他の例を示す。このサイズに対しては、最初の3つの変換レベルは、142 10 ×128バッファ内のメモリを再使用できる。

【0045】

【表2】

表2

レベル	別のメモリ (バイト)	再使用されたメモリ (バイト)
1	128×128 = 16,384	2×128 = 256
2	64×64 = 4,096	4×128 = 512
3	32×32 = 1,024	8×128 = 1024

一実施例では、適切なメモリ又は新たなメモリを使用する判断は、タイルの高さと変換レベルの関数である。そのような判断は、以下に基づいている。タイル高さ>2 (3*level-2) の場合には、適切な方法を使用する。タイル高さ=2 (3*level-2) の場合には、いずれかを使用する。タイル高さ<2 (3*level-2) の場合には、新たなメモリを使用する。

【0046】判断の応用を示すために、表3は以下の様である。

【0047】

【表3】

表3

レベル	2^(3*level-2)
1	2
2	16
3	128
4	1024
5	8192

幾つかの応用では、メモリ構成をタイル高さに適応させることは不便である。単一の固定のメモリ構成が使用できる。128×128より小さいタイルサイズは、典型的には、好ましくない圧縮性能となり、従って、典型的には、使用されない。しかし、1K×1Kよりも大きなタイルサイズが非常に大きな画像に対して使用される場合には、これは、圧縮を大きくは改善せず、そして、必要とされる大量のメモリは、典型的には、負担が重い。従って、128から1024の間のタイル高さを仮定し、そして変換の3レベルに対して適切なメモリを使用することが、実際には良い。

【0048】復号は同様であり、復号論理が読む前に、

逆変換を与えた結果が書きこまれ、顕著な差は開始が最高レベルから最低レベルへ、上述の例ではレベル6からレベル1に開始されることのみである。そのような場合には、入力タイルは、メモリ構造の一番上で終わる。メモリの再使用を収容する余分なラインは、降順である。例えば、図4(B)の構造を使用して、L3係数からL2係数を形成するのに8ラインが必要であり、L2係数からL1係数を形成するのに4つの余分なラインが必要であり、そして、L1係数から入力タイルを形成するのに2つの余分なラインが必要である。

【0049】一実施例では、入力タイルデータを扱うために、符号化に先だって色変換が行われ得る。図5は入力データを扱う処理の一実施例を示す。図5を参照すると、カラー入力画素が、ラスタ順序で受信される。これらのカラー画素は、RGB、YCrCb、CMY、CMYK、グレースケール等である。カラー入力画素は、バンド(又は、他の形式)により、メモリ501のような、メモリ内にタイルとして記憶されても良い。

【0050】段階501からの又は入力から直接受信された画素は、色変換及び/又はレベルシフトを受け、結果の出力は1つの係数バッファ502₁ - 502_Nに記憶される。即ち、各タイルに対して一旦色変換が完了すると、1つの係数バッファ502₁ - 502_Nに記憶され、そして、次のタイルが処理される。一実施例では、各成分に対して1つの係数バッファがある。

【0051】係数バッファ502₁ - 502_Nは、メモリを再使用しながらウェーブレット変換を行うために上述したように、変換で使用される。従って、係数バッファ502₁ - 502_Nは、入力と出力ともにウェーブレット変換である。

【0052】変換が、係数バッファ502₁ - 502_Nに与えられた後に、関係モデル503とエントロピー符号化器504は、既に変換されたデータについて、更なる圧縮処理を実行する。符号化されたデータは、符号化データメモリ505内にバッファされる。

【0053】1つのタイルに更なる圧縮処理を行いながら、変換は他のタイルに適用されてもよい。同様に、どの又は全ての動作は、同時に複数のタイルに行われても良い。

【0054】進行順序変換

JPEG2000規格では、圧縮されたコードストリーム内のデータは、5つの進行順序の1つに記憶される。進行順序は、コードストリーム内の異なる点で変わり得る。順序は、階層、区画、解像度及び、成分に関する、埋め込まれた"for layers(階層)"により、定義される。

【0055】5つの進行順序がJPEG2000規格のテーブルA-16内の規格内に記載されている。それらは、階層-解像度-成分-位置進行(LRCP)、解像度-階層-成分-位置進行(RLCP)、解像度-位置

一成分一階層進行(RPCL), 位置一成分一解像度一階層進行(PCRL), 成分一位置一解像度一階層進行(CPRL)である。

【0056】順序は、JPEG2000規格内のCOD又はPODマーカー内で定義される。符号化形式デフォルト(COD)マーカーはJPEG2000規格で定義されそして、符号化形式、分解レベル、及び、(メインヘッダ内であれば)画像又は(タイル部分ヘッダ内であれば)タイルの全成分を圧縮するのに使用されるデフォルトである階層化を記述する。進行順序変更(POD)マーカーは、コードストリーム内のCODマーカーセグメント内に規定されている以外のどの進行順序に対しても境界と進行順序を記述する。パケット長メインヘッダ(PLM)は、順に各タイル部分に対してタイル部分内のパケット長のリストとパケット長を示し、タイルパートヘッダ(PLT)は、タイル部分内及び、タイルパケット長を示し、且つコードストリーム内にデータがある場所を示す。

【0057】B. 12章内のJPEG2000規格は、所定の進行順序に対して、圧縮データのパケットがどのような構成されるかのみを規定する。1つの進行順序から他の進行順序へデータがどのように変換されるかは記述しない。

【0058】一実施例では、進行順序変換パーサは、データ復号しそれを再度符号化すること無しに、ユーザの入力に基づいて、コードストリームを所望の進行順序に変換する。図6(A)は、そのようなパーサを有するシステムを示す。図6(A)を参照すると、パーサ601は、特定の進行順序に関するクライアントからの要求を受信する。このクライアントはウェブページを見ながら、特定のリンクを選択する。この要求に応答して、パーサ601は、サーバ602にアクセスし、メモリ604から全画像603に関連するコードストリーム得て、そして、要求に基づき、コードストリームを異なる進行順序に変更する。この要求は、オプションコマンドを使用して(例えば、RL2L(解像度一階層進行から階層進行へ)進行順序を示す。記述されている進行順序は、階層、解像度、成分、区画又は、タイルに基づいている。

【0059】図6(B)は、階層進行コードストリーム(LRCP)から解像度進行(RLCP)コードストリームへ変換する進行変換器を示す。進行順序は、互いに直接的にマップする。

【0060】図7(A)は、1つの進行順序から他の進行順序へコードストリームを変換する複数の経路を示す。図7(A)を参照すると、5つの進行順序(LRCP, RLCP, RPCL, CPRL及び、PCRL)の各々は、互いへの経路と共に示されており、これにより全ての進行が示されている。一実施例では、パーサは、全ての交換に、階層進行を最初に通し、そして、選

択された変換を行う。図7(B)は、そのような単純化された変換経路の一実施例を示し、要求されるマッピングの数は、(図7(A)に示されているように)10から4へ減少される。しかしながら、5つの進行順序の内のどの1つも、選択された順序に到達する前に全てが変換される1つとして使用されることが可能である。ここで説明した変換技術は、ソースコードのラインの数が多数の経路の変換よりも非常に少なくなるように、ソースコードを単純化する。これは、デバック時間を減少させ、且つ少ないメモリと実行時変数という結果となる。

【0061】変換を実行するために、コードストリーム内のパケットの順序がリオーダーされなければならない。パケットは、コードストリーム内でその連続順序によりラベルが付される。マーカーは、データの開始点、データの長さ(又は、代わりに、データの終了点)及び、どのようにデータが扱われるべきかを示す。例えば、どのようにデータが扱われるべきかの指示は、データが削除されるべきか否か、データが切り捨てられるべきか否か、又は、データに他の動作が行われるべきかを示す。そのような取扱情報は、JPEG2000規格のPLT/PLM及び/又はPPT/PPMマーカーセット内に設けられているような、レート歪情報からも来る。この方法で、コードストリームはパケットヘッダを変更することなく切り捨てることが可能である。

【0062】一実施例では、リスト、配列又は、(601Aのリオーダーリング構造のような)他の構造は、各パケット内のデータの部分を示すことにより構築される。この構造を使用して、パケットはリオーダーされる。

【0063】図8は、進行順序変換を実行する処理の一実施例を示す。この処理は、ハードウェア(例えば、専用の論理、回路等)、(汎用コンピュータ又は、専用機械で実行される)ソフトウェア、又は、その両者の組合せを含む処理論理により行われる。

【0064】図8を参照すると、処理は、パケット内のヘッダからのリストを構築する処理論理により開始し(処理ブロック801)、そして、随意に、マーキングリスト項目を、量子化のために"消去"する(処理ブロック802)。次に、処理論理は、元の進行を、(入力と出力をPOCマーカー(進行順序に関する境界)で規定された進行で扱うことを含む)所望の進行にマップするためにリストのリオーダーを行う(処理ブロック803)。その後、処理論理は、リオーダーリストに基づいて符号化されたデータを出力する(処理ブロック804)。

【0065】従って、リオーダーと解析の組合せは、所望のリオーダーリング及び、解像度、品質等の仕様を可能とする。

【0066】進行順序変換の例

以下は、コードストリーム内で、どのようにパケットが配置されるかを示す例である。コードストリームは、2

10

20

30

40

50

成分、2階層、3分解レベル及び、階層進行に基づいて、形成されている。

【0067】表4は、パケット順序、長さ及び、この例内のパケットの関連インデックスを示す。パケット順序列は、コードストリーム内に配置されたパケットの連続順序を示す。長さは、パケットの長さを示す。関連インデックスは、解像度、階層、成分及び、パケットの区画を示す。

*

表4

パケット順序	長さ	関連インデックス
packet[0]	length=589	R0L0C0P0
packet[1]	length=589	R0L0C1P0
packet[2]	length=924	R1L0C0P0
packet[3]	length=924	R1L0C1P0
packet[4]	length=1602	R2L0C0P0
packet[5]	length=1602	R2L0C1P0
packet[6]	length=733	R3L0C0P0
packet[7]	length=733	R3L0C0P0
packet[8]	length=535	R0L1C0P0
packet[9]	length=535	R0L1C1P0
packet[10]	length=1523	R1L1C0P0
packet[11]	length=1523	R1L1C1P0
packet[12]	length=5422	R2L1C0P0
packet[13]	length=5422	R2L1C1P0
packet[14]	length=16468	R3L1C0P0
packet[15]	length=16468	R3L1C1P0

このコードストリーム内では、パケットは、それらがある階層に基づいてグループ化されている。最初の8パケットは階層0に属する。続く8パケットは階層1に属する。

【0070】ここで説明した変換処理を使用して、上述のコードストリームは解像度階層進行に変換される。以下はどのように上述のパケットがリオーダーされるかを示す。

*

表5

前のパケット順序	パケット順序	長さ	関連インデックス
0	packet[0]	length=589	R0L0C0P0
1	packet[1]	length=589	R0L0C1P0
8	packet[2]	length=535	R0L1C0P0
9	packet[3]	length=535	R0L1C1P0
2	packet[4]	length=924	R1L0C0P0
3	packet[5]	length=924	R1L0C1P0
10	packet[6]	length=1523	R1L1C0P0
11	packet[7]	length=1523	R1L1C1P0
4	packet[8]	length=1602	R2L0C0P0
5	packet[9]	length=1602	R2L0C1P0
12	packet[10]	length=5422	R2L1C0P0
13	packet[11]	length=5422	R2L1C1P0
6	packet[12]	length=733	R3L0C0P0
7	packet[13]	length=733	R3L0C1P0
14	packet[14]	length=16468	R3L1C0P0
15	packet[15]	length=16468	R3L1C1P0

【0068】例えば、パケット[0]は、第1のタイルヘッダ後のコードストリーム内の最初のパケットである。それは、589バイトを有する。関連インデックスRwLxCyPzは、解像度w、階層x、成分y及び、区画zに即するパケットを示す。

【0069】

【表4】

【0071】階層進行コードストリームが解像度進行に変換された後に、新たなコードストリーム内で、パケットは解像度に基づいてグループ化される。そのようなグループ化を表5に示す。最初の4パケットは解像度0に属し、次の4パケットは解像度1に属する等である。

【0072】

【表5】

【0073】

* * 【数2】

```

n = 0;
for(l=0;l<layer;l++){
    for(r=0;r<resolution+1;r++){
        for(c=0;c<component;c++){
            new_packet[n] = old_packet[l*component + r*layer*component +
c];
            n++;
        }
    }
}

```

Layer to Resolution Progression

```

n = 0;
for(r=0;r<resolution+1;r++){
    for(l=0;l<layer;l++){
        for(c=0;c<component;c++){
            new_packet[n] = old_packet[r*component +
l*(resolution+1)*component + c];
            n++;
        }
    }
}

```

ここで、*layer*=コードストリーム内の階層の番号、*resolution*=コードストリーム内の分解レベルの番号、そして、*component*=コードストリーム内の成分の番号である。

【0074】JPEGL2000符号化のにデータ隠蔽(サイドバンド情報)

ビット隠蔽は、ファイルサイズを増加させずにサイドバンド情報を伝送することを可能とする。(JPEGL2000で定義されているCOMマーカが代わりに使用され得るが) ファイルサイズを増加するがしかし、単純な復号器を遮断しないサイドバンド情報も重要である。

【0075】幾つかのマーカセグメント、パケットヘッダ及び、パケットは、元も近いバイトにパディングされる。JPEGL2000マーカセグメントの例は、PPM、PPT、PLM及び、PLTである。更に加えて、幾つかのマーカセグメントは、QCD、QCC及び、POCを含むのに必要なよりも長くてもよい。全てのこれらの場合において、パディングデータ値は定義されていない。

【0076】幾つかの専用の符号化機構は、この半ランダムに配置された未定義のデータを、制限はされないが、復号及びフィルタリング指示、所有権、セグメント化指示等を含む、幾つかの重要な形式の情報を提供するために使用できる。指示は、特定の改善機構に対するインデックスを含んでも良い。例えば、画像がほとんどテキストであると知られている場合には、第1の前処理フ

ィルタが使用されるべきであるということを示す値が送られる。一方では、領域が大部分グラフィック画像である場合には、第2の前処理フィルタが使用されるべきであるということを示す値が送られる。

【0077】以下は、コードストリーム内にビットが隠される又はサイドバンド情報が蓄積され得る場所である。

- (予測可能な終点無し)の算術符号化器(AC)終点、
- バイトに丸められるパケットヘッダの最後、
- 最後のパケットの後で、次のタイルの前、
- 常に最小を使用しないことによるタグツリー構造、
- パケットヘッダLブロック信号、
- コードブロックに対するLSBパリティ(改良パスのみ、クリーンアップパスのみ)、
- QCD、QCC余分サブバンド、POCである。

【0078】例えば、AC終点を使用するデータの隠蔽に関して、少なくとも、符号化器が終端される度に、0から7ビットが設けられる。しかしながら、これは、幾つかのバイトに拡張される。これらの特別なビットとバイトは特別な情報を送るのに使用される。

【0079】各パケットヘッダに関して、パケットヘッダの最後はバイト境界に丸められる。従って、丸めが必要な度に特別な情報を送るために1から7ビットが使用される。同様に、各パケットは、バイト境界に丸められ、それにより、1から7ビットを設ける(丸めは必要

30

40

50

19

であると仮定する)。また、タイル部分内の最後のパケットは、数バイト拡張されることが可能である。これらの特別なバイトは、追加及び、情報を送るのに使用される。

【0080】コードブロックに対する圧縮されたデータの長さは、最小でない表現のパケットヘッダ内に与えられることが可能である。表現の選択（例えば、非最小表現）は、他の情報を指示するために使用できる。

【0081】タグツリーデータの隠蔽に関して、JPEG2000規格のパケットヘッダは、第1の包含とゼロビット面情報の符号化のためにタグツリーを使用する。複数のコードブロックがある場合には、タグツリーは最小値の4つのツリーのようである。例えば、パケット内の4x4配置内の16コードブロックの場合には、配置は以下の様である。

【0082】

【数3】

```

10 7 12 15
3 20 21 5
81 45 5 9
18 8 12 24

```

上述の4x4配置に対して最小のである、タグツリーの例は、次のようである。

【0083】

【数4】

```

3 0 2 7 4 7 10
5 2 0 17 16 0
73 37 0 4
10 0 7 19

```

ここで、“3”は各コードブロックの値に加算されそして、“0”、“2”、“5”及び、“2”は、各々が4つの対応するコードブロックに加算される。最後に、コードブロック毎に1つの値がある。即ち、最小のタグツリーは、上述の4x4配置内の最初の2x2グループを取ることに形成されそして、最小値を見つけることは、4つの値の中からである。この場合には、2x2ブロックに対して、

【0084】

【数5】

```

10 7
3 20

```

であり、最小値は3である。これは、他の2x2ブロックに対しても行われる。そして、これらの識別された最小値は、この例では“3”である、それらの最小値を決定するために再び評価される。そして、最小値は4つの最小値から減算され、以下の

【0085】

【数6】

```

0 2
5 2

```

20

を形成する。そして、4x4内の残りの数に対して、数3が、4x4配置内の特定の値に対応する、2x2内の値と共に各値から減算され、それにより、上述のタグツリーとなる。最初の行は以下のように加算される。

10=3+0+7

7=3+0+4

12=3+2+7

15=3+2+10

可変長コードが、小さな数を表現するのに効率的に使用され得る。

【0086】最小でないタグツリーの例は、次のようである。

【0087】

【数7】

```

2 1 3 7 4 7 10
6 3 0 17 16 0
73 37 0 4
10 0 7 19

```

(“3”、“0”、“2”、“5”及び、“2”を表すことは、“2”、“1”、“3”、“6”及び、“3”よりも少ないビットストリームデータを使用し得ることに注意する。)

一旦、タグツリー表現が作られると、2x2ブロック内にゼロがあるか否かに基づいて、表現が最小か否かに関する決定を行うことができる。従って、この情報は隠される。例えば、1ビットブロックは、2x2ブロック内の1を示し、上述は、最小タグツリーの一部ではないことを示すが、しかし幾つかの特定の情報を復号器に送るために使用できる。同様に、2が2x2ブロック内で最小であれば、そのような事実は復号器に異なる情報を送りうる。

【0088】JPEG2000 POC、QCD及び、QCCマーカは、冗長なエントリを有することが可能である。あたかも、コードストリームは量子化されそして、マーカは再書き込みされていないようである。例えば、QCDとQCCマーカは、マーカの構文により規定される幾つかのサブバンドに対する値を有する。ビットストリーム内により少ないサブバンドが実際に符号化されている場合には、データは、サブバンドを見失わさせるために使用される値内に隠され得る。冗長なエントリは、置換されそして、隠された又はサイドバンド情報のために使用されうる。

【0089】隠された又はサイドバンド情報は、（例えば、特定のフィルタでそのタイルをシャープ化する又は、強調又は、平滑化又は、この領域に光学的文字認識(OCR)を実行する等の）後処理指示、復号指示、（例えば、残りの画像又は他の画像を復号するための暗号鍵等の）セキュリティ、（例えば、ファイルのオリジネータとしてのPOTUSラベル付け等の）コードストリーム識別及び／又は他の情報を含んでも良い。

【0090】符号化時の階層の使用

階層は、J P E G規格の一部である。一実施例では、おそらくCOMマーカのサイドバンド情報は、復号中に階層の選択を可能とするために、復号器により使用される。このサイドバンド情報は、異なる視聴距離、異なる解像度、異なる興味の領域、（例えば、テキストのエッジを見つけるような）分析のための異なる周波数内容に対する、レート／歪目標に合致させるために、圧縮後量子化のための階層を選択するのに使用される。

【0091】一実施例では、階層はレートに基づいて予め定義される。例えば、第1の階層は、画素画像当り1ービットを表し、一方、第2の階層は、画素画像当り2ービットを表すというようである。従って、階層は、最

も低い品質から最も高い品質へと変わる。同様に、目標レートも、低解像度に合わせることができる。

【0092】サイドバンド情報は、コードストリーム内のマーカセグメントに蓄積される。一実施例では、J P E G 2 0 0 0 コメント（COM）マーカは、階層に関する情報を提供するように使用される。特にCOMマーカは、各解像度及び/又は全体画像にわたるレートに関するバイト数又は、各追加の階層に対する相対的なバイト数を示すのに使用される。表6は、画像内のタイルをわたり、バイト数で、各層とその解像度を示す。そのようなテーブルは代わりに歪値を有する。

【0093】

【表6】

表6

lev=0	layer=0	comp=0	bytes=529
lev=0	layer=0	comp=1	bytes=555
lev=0	layer=0	comp=2	bytes=493
lev=0	layer=1	comp=0	bytes=129
lev=0	layer=1	comp=1	bytes=130
lev=0	layer=1	comp=2	bytes=123
lev=0	layer=2	comp=0	bytes=7
lev=0	layer=2	comp=1	bytes=8
lev=0	layer=2	comp=2	bytes=12
lev=0	layer=3	comp=0	bytes=1
lev=0	layer=3	comp=1	bytes=1
lev=0	layer=3	comp=2	bytes=129
lev=1	layer=0	comp=0	bytes=705
lev=1	layer=0	comp=1	bytes=898
lev=1	layer=0	comp=2	bytes=712
lev=1	layer=1	comp=0	bytes=146
lev=1	layer=1	comp=1	bytes=114
lev=1	layer=1	comp=2	bytes=116
lev=1	layer=2	comp=0	bytes=224
lev=1	layer=2	comp=1	bytes=250
lev=1	layer=2	comp=2	bytes=263
lev=1	layer=3	comp=0	bytes=201
lev=1	layer=3	comp=1	bytes=212
lev=1	layer=3	comp=2	bytes=200
lev=2	layer=0	comp=0	bytes=889
lev=2	layer=0	comp=1	bytes=1332
lev=2	layer=0	comp=2	bytes=1048
lev=2	layer=1	comp=0	bytes=240
lev=2	layer=1	comp=1	bytes=329
lev=2	layer=1	comp=2	bytes=328
lev=2	layer=2	comp=0	bytes=599
lev=2	layer=2	comp=1	bytes=767
lev=2	layer=2	comp=2	bytes=725
lev=2	layer=3	comp=0	bytes=335
lev=2	layer=3	comp=1	bytes=396
lev=2	layer=3	comp=2	bytes=420
lev=3	layer=0	comp=0	bytes=1
lev=3	layer=0	comp=1	bytes=395
lev=3	layer=0	comp=2	bytes=402
lev=3	layer=1	comp=0	bytes=251
lev=3	layer=1	comp=1	bytes=450
lev=3	layer=1	comp=2	bytes=562
lev=3	layer=2	comp=0	bytes=525
lev=3	layer=2	comp=1	bytes=990
lev=3	layer=2	comp=2	bytes=1313
lev=3	layer=3	comp=0	bytes=1214
lev=3	layer=3	comp=1	bytes=1798
lev=3	layer=3	comp=2	bytes=2585

他の実施例では、整列は、階層毎である。このように、上述の情報は、次の

階層=0、バイト=7959、ビットレート=0.971558、PSNR=30.7785による整列、
 階層=1、バイト=10877、ビットレート=1.327759、PSNR=32.0779による整列、
 階層=2、バイト=16560、ビットレート=2.021484、PSNR=35.7321による整列、

に示すように、各レベル（レベル又は成分により分離されていない）に対して統合される。

【0094】階層による歪は、PSNRに基づいている。例えば、

階層=0、PSNR=30.7785
 階層=1、PSNR=32.0779
 階層=2、PSNR=35.7321
 である。

【0095】代わりの実施例では、そのような情報は、上述のようにコードストリーム内に隠されている。この情報は、レート歪を制御するのに使用され得る。

【0096】他の実施例では、階層は、特定の視聴距離で予め定義される。そのような場合には、データは、最も高い周波数、最も低い解像度から、最も低い周波数、最も高い解像度へ、階層に分割される。

【0097】一実施例では、階層情報は、その階層と全ての前の階層に対する全体画像をわたるビットの合計を示す。(例えば、階層1の次にリストされている16,011ビットは、階層0と階層1に対する全ビット数を示す。)表7は、この種の絶対レート情報を示す。

【0098】表8は、相対的なレート情報を示す。階層0は4096ビット、階層1は11,915ビット等を示す。

【0099】

【表7】

表7

階層	レート(バイト)
0	4,096
1	16,011
2	40,000
3	100,000
4	250,000
5	500,000
6	1,000,000
7	2,500,000
8	5,500,000

【0100】

【表8】

表8

階層	レート(バイト)
0	4,096
1	11,915
2	23,989
3	60,000
4	150,000
5	250,000
6	500,000
7	1,500,000
8	3,000,000

例えば、750,000バイトのみが画像を復号するのに許されている場合には、復号されうる全ては、(階層6に示されている1,000,000バイトは、階層0-5の500,000バイトを含むので)、階層5までと、重要な階層6の半分である。幾つかの実施例では、階層6からのパケットは含まれない。他の実施例では、階層6データの合計が約250,000バイトとなるように、階層6からの幾つかのパケットが含まれそして、その他のパケットは、ゼロパケットに置き換えられ

る。

【0101】図22は、3レベル、MSE又は、同様な5.3不可逆変換に対する階層化の例を示す。図22を参照すると、45階層が示されている。各追加の階層はMSEに対して、良好なレート歪を与える順序でMSEを改善する。

【0102】図23は、変換が5レベルを有し且つデータ階層0-3にまで分割されている他の例を示す。階層0はサムネール版に対応し、階層0-1はモニタ(又は、スクリーン)解像度に対応し、階層0-2は印刷解像度に対応しそして、階層0-3は無損失に対応する。

【0103】代わりの実施例では、階層は幾つかの他の歪基準(例えば、MSE、重み付けされたMSE、テキストの鮮明さ等)で予め定義されても良い。

【0104】復号器は、画像を発生するために階層を選択するために、コードストリームから階層に関する情報を使用する。アプリケーション又は実行から望ましい視聴環境(表9参照)を知っており、且つ、階層を規定すコードストリームからの情報を使用する復号器は、但し視聴距離で画像を表示するために、コードストリームを量子化できる。図9はそのような復号器を示す。図9を参照すると、復号器901はコードストリームを受信し且つ、COMマーカを検査する量子化論理902を含み、且つ、例えば、適切な階層を選択することを介して、量子化されたコードストリーム904を発生するために、メモリ903内に記憶された視聴距離に関する情報を使用する。量子化されたコードストリーム904は、画像データ906を発生するために階層を選択した後、復号論理905(例えば、JPEG2000復号器)により復号される。単純な復号器は、コメントマーカ内のデータを単に無視する。

【0105】図10は、復号時に階層を使用するための処理のフローを示す。この処理は、ハードウェア(例えば、専用の論理、回路等)、(汎用コンピュータ又は、専用機械で実行される)ソフトウェア、又は、その両者の組合せを含む処理論理により行われる。

【0106】図10を参照すると、この処理は、圧縮された論理データのコードストリームを受信する処理論理により開始する(処理ブロック1001)。この画像データは、複数の階層に組織化され、その各々は、画像に(例えば、シャープに見える、より明瞭である、よりコントラストがよい等の)、視覚値を画像に加える符号化されたデータを含む。次の処理論理は、サイドバンド情報に基づいて量子化のために1つ又はそれ以上の階層を選択する(処理ブロック1002)。選択後に、処理論理は、コードストリームの量子化されていない階層を逆圧縮する(処理ブロック1003)。

【0107】タイル、タイルの一部分及び、パケットの編集

一旦、コードストリームが形成されると、画像の一部を

編集することが望ましい。即ち、例えば、コードストリームを形成するために、符号化を行った後に、タイルの組が復号される。タイルの組を復号した後に、編集が行われ、それに続いて、復号される前に符号化されていたタイルと同じサイズへ、編集されたものと共にタイルの組みが符号化される。典型的な編集の例は、テキストの「シャープ化」と「赤目」の除去を含む。J P E G 2 0 0 0 コードストリームは、全体のコードストリームを書き直すこと無しに、メモリ内で又は、ディスクファイルシステム内で編集されることが可能である。

【0108】図11は、編集処理の一実施例のフローを示す。この処理は、ハードウェア（例えば、専用の論理、回路等）、（汎用コンピュータ又は、専用機械で実行される）ソフトウェア、又は、その両者の組合せを含む処理論理により行われる。

【0109】図11を参照すると、処理論理は最初に、編集されるべき領域、解像度、成分及び／又は区画を包含する、タイル、タイル部分又は、パケットを決定しそして、それらを復号する（処理ブロック1101）。この決定は、領域及び／又は動作解像度を選択するユーザに20 応答して行われる。この決定は、どの部分又はタイルが編集される部分を包含するかを決定するために、より高い解像度によりする編集情報を使用しても良い。一旦復号が完了すると、処理論理は、所望の編集を実行する（処理ブロック1102）。

【0110】所望の編集を実行した後、処理論理はデータを、符号化されたデータに再圧縮し（処理ブロック1103）、そして、置換タイル、タイル部分又は、コードストリームに対するパケットを形成する（処理ブロック1104）。一実施例では、置換タイル、タイル部分又は、パケットを形成する際に、処理論理は、新たなデータが、データの未編集版よりも小さい場合には、未編集版と同じサイズの置換タイル、タイル部分又は、パケットを形成するために、コードストリームの最後にバイトのデータをパディングする。

【0111】代わりの実施例では、処理論理は、パディングの代わりに、適切な長さのCOMマーカーセグメントのような、マーカー又はタグを使用しても良い。COMマーカーは、スペースを埋めるために使用でき又は、符号化器が含めたい情報を含むことができる。例えば、ここで説明したサイドバンド情報又は、画像に関する著作権ライセンス又は、テキスト又は、他のフィルフォーマット情報のような情報を含むべきである。

【0112】一実施例では、置換タイル、タイル部分又は、パケットの形成に際し、新たなデータがデータの未編集版よりも大きい場合には、処理論理は、データがコードストリーム内に合うまで、どの又は全ての成分を切り捨てる。

【0113】画像の編集は、タイル、タイル部分又は、コードブロックに対する符号化されたデータを変更する

ことにより行われる。一実施例では、編集は、拡張する代わりに量子化することによりファイルを変更せずに行われる。他の実施例では、所定の量の拡張を可能とするために、所定の量の更なる空間が、タイル又は、コードブロック当りに割当てられる。更に他の実施例では、符号化されたデータは、タイルヘッダを操作することによりファイルの最後に置かれそして、COMマーカー内に無効なタイルデータを配置する。

【0114】編集されているコードストリームの部分内のデータに依存する後続のタイル部分がある場合には、これらのタイル部分はコードストリーム内で使用できなくなることに、注意する。この使用できないデータの指示は、幾つかの方法の内の1つにより復号器に知らせられる。これらの方法は、使用できないデータの存在及び/又は位置を示すために、情報を挿入し又は変更することを含む。1つの実施例では、アプリケーションは、編集されたタイル部分に後続するタイル部分内のデータが使用できない可能性があることを示すために、状態バッファを使用する。状態バッファは、ワークメモリ内であっても良くそして、パケットの間の依存関係を記述する。以前のパケットが変更された場合には、後続のパケットはそのまま復号されることはできない。これらの後続のパケットは、それに応じて編集されるか又は、削除されねばならない。他の実施例では、そのような指示はそれらのタイル部分のデータ部分をゼロにするか及び/又は、データがないことを示すPPTマーカーを生成することによりなされてもよい。

【0115】最適な符号化器量子化

符号化中に、幾つかの又は全てのサブバンドからの量子化のされていない係数は、量子化された係数値を生成するために、Q値により除算される。このQ値は広範囲な値を有する。典型的な符号化器は、1つの単一の係数値に等しくされる、単一の範囲の値内の幾つかの値を量子化する。本質的には、特定の範囲内の全ての係数は、同じ値に量子化される。これは、図12により例示されており、値の範囲は、釣鐘状の形状の曲線を有しそして、範囲R₁のような特定の範囲内の全ての値は、R₁'のような1つの量子化された値として復号器に送られ、そして、復号器は、これらの値を特定の値に再構成する。復号器はこれらの値を（ここで、Qが量子化ステップサイズであるとする、例えば、 $\text{floor} (1/2 \cdot \min + 1/2 \cdot \max)$ 又は、 $\min + 1/2 \cdot Q$ のように、）所定の値に再構成すると仮定する。例えば、値の範囲が16と31の間にある場合には、復号器は値が24であるとみなしうる。1つの実施例では、値として1/2を使用する代わりに、 $\text{floor} (3/8 \cdot \min + 5/8 \cdot \max)$ 又は、 $\min + 3/8 \cdot Q$ のような、他の値が選択される。ここで、Qが量子化ステップサイズである。従って、範囲が16と31の間である場合には、復号器は、その値を、24の代わりに22に再

構成する。

【0116】幾つかの場合には、2つの空間的に隣接する係数は、互いに数値的に近いが、図12において、範囲R₂の係数値1201と範囲R₁の係数値1202のような、別の量子化ビン（量子化範囲）内にある場合がある。この量子化の結果では、歪が起り得る。一実施例では、2つの量子化ビン（量子化範囲）の境界付近にある係数に対しては、符号化器は、係数1201のような係数が量子化される範囲R₁のようなビン（範囲）を選択し、これにより、係数1202のような隣接する係数との間の一貫性が保たれる。これは、歪を避けるのに役立つ。即ち、この技術は、特に係数が小さいビン（範囲）から高いビン（範囲）に移動したときに、歪を減少し、そして、レートを増加させる。

【0117】モーションJ P E Gに対するフリッカの低減

動画シーケンスにウェーブレット圧縮を適用する場合には、時々、フリッカが発生する。そのようなフリッカの例は、動画シーケンスが再生されるにつれて、連続するフレーム内の、領域内が明るくなる又は暗くなる画像やエッジ変化の出現（エッジ付近のモスキートノイズ）を含みうる。そのフリッカは、動画シーケンスの連続するフレームに対する又は、一時的に視聴される量子化により悪化される雑音に対する、異なった局所的な量子化の適用に依っている場合がある。

【0118】フリッカを減少させるために、連続するフレーム内で、同じ位置内にある及び、同じ値に近い係数は、同じ値になされる。即ち、連続するフレーム内の係数値は、所定の値に設定される。これは、本質的に、符号化中に適用される量子化の形式である。図13は、フリッカを減少させる処理の一実施例のフロー図である。

【0119】後続するフレーム内の係数値にそのような量子化を適用するか否かの試験は、前のフレーム内の係数に行われた量子化に基づいている。従って、復号器がフレームごとに独立にデータを復号しながら、符号化器は、フリッカを除去するためにフレームの依存性を利用する。

【0120】一実施例では、モーションJ P E G内のフリッカを減少させるために、係数値は、しきい値に関する互いの関係に基づいて、変更され（量子化され）る。例えば、D_nとD_{n+1}が、量子化前の2つのフレーム内の対応する係数（同じ空間位置と同じサブバンド）である場合には、D'_nとD'_{n+1}が、量子化後のこれらの係数を表す場合には、Q（・）がスカラー量子化の場合には、そして、値Tがしきい値である場合には、以下の、

10

である。

【0122】係数値の1つは、他の係数値へのいずれかの所定の近似となるように変更されてもよい。この近似はあるしきい値により決定される。しきい値は、ある基準に基づいて、ユーザが設定するか又は適応的である。このしきい値は、サブバンドと、おそらく、特定の値の一貫性（この係数が近いフレームの数）に基づいて異なっても良い。一実施例では、係数値は他の係数値と等しく設定される。代わりの実施例では、係数は、他の係数値の量子化ビン（量子化範囲）サイズ内に又は、量子化ビン（量子化範囲）サイズの2倍に設定される。

【0123】図14は、上述の量子化を実行する符号化器（又は、その部分）の一実施例を示す。図14を参照すると、量子化器1400は、（図示していない）ウェーブレット変換からの動画シーケンスのフレームに対する係数1410を受信する。その係数は、量子化論理1401により受信される。係数1410に対してメモリ1403内に記憶された前のフレームに関する係数値に対してメモリ1403内に記憶されたしきい値と、メモリ1404から与えられるスカラー量子化値Qとを比較する。

【0124】量子化論理1401は、（例えば、ゲートを有する論理、回路等のような）比較ハードウェア又は、比較を実行するソフトウェアを含み得る。この比較ハードウェアとソフトウェアは、減算器又は減算動作を実行する。この結果は、コードストリームに量子化される（いくつかの値が変更されたとする）。

【0125】これは、2つ又はそれ以上のフレームにわたって適用される。また、その比較は2つの連続するフレームに制限されない。比較は、例えば、変化が存在するかどうかを決定するために、3、4、5等のフレームをわたっても良い。図24は、第1と第3のフレーム内の値が、第2のフレーム内の値を設定するのに使用される例を示す。

【0126】量子化は、コードブロックに基づく規則を用いるコードストリーム量子化でもよいことに注意する。

【0127】レート制御、量子化及び、階層化
一実施例では、更に可能性の高いシンボル（MPS）であるべき、改良ビットのサブセットを設定することによ

50

if (|Q (D_{n+1}) - (D' n)| < T)

D' n+1 = D' n

else

D' n+1 = Q (D_{n+1})

が適用される。例えば、値Tは、量子化ステップサイズ
の2倍でもよい。Tの他の値は、制限はされないが、

【0121】

【外1】

$$\sqrt{2}Q, 1.5Q, 2\sqrt{2}Q$$

30

り、係数の選択的量子化が、符号化中に行われる。これは、ユーザ選択されたビット面で行われてもよい。例えば、背景画像にテキストがある場合には、背景画像に対して要求される符号化されたデータを最小化しながら、シャープなテキスト画像を有することを目的として、MPSに設定された改良ビットは、テキストに影響を与えるビットに対する実際の値を使用しながら、最後のビット面に対するテキストに影響を与えないものである。

【0128】そのような量子化機構は、不均一な量子化サイズを実行するのに使用される。例えば、少ないビットでの背景を有することを望む場合には、MPSへの改良ビットの設定は、量子化の形式として動作できる。この量子化機構は、あるレベルの歪を発生するが、コードストリームを伝送するのに必要なビットレートを下げる。

【0129】この技術は、改良パスの間に発生されたビットに与えられうるが、この技術は他の圧縮機構（例えば、下位のパスの間に発生されるリスト、カリフォルニアのメンロパーク（Menlo Park）のリコーシリコンバレーのCREWのテールビット、MPEG I 20 V テクスチャーモード等）への応用を有することに、注意する。

【0130】一実施例では、同じ技術が、フレーム間の他の変化に適用されてもよい。即ち、一実施例では、1フレーム内のレート歪による変化は、歪効果を防ぐために、後続のフレーム内で行われる。

【0131】レート制御と量子化

一実施例では、ユーザの規定する量子化が提供される。1つの成分の3レベル変換に対しては、レベル1HH、レベル1HL及びLH、レベル2HH、レベル2HL及びLH、レベル3HH、レベル3HL及びLH、及び、レベル3LHの、7の量子化値で十分である。

【0132】量子化値が、（2のべき乗によるスカラー量子化に等しい）切り捨てるビット面である場合には、大部分の応用では、3-ビット値（0...7）で十分である。（12ビット又はそれ以上のビット深さと5又はそれ以上の変換レベルを持つ画像成分に対しては、おそらくより高い量子化が有益である。）値0...6は、切り捨てるビット面の数を規定するのに使用され、そして、7は全てのビット面を捨てることを意味するの40 に使用される。3ビット値は、量子化を実行するために、圧縮（又は、逆圧縮）ハードウェア（例えば、JPEG 2000互換ハードウェア）を制御する、コントローラに書きこまれる。

【0133】3成分カラー量子化に対しては、-21値を、各成分に対する別々の値とともに使用でき、-輝度に対して7且つ、色差に対して7の、14値を、使用でき、-輝度に対して7且つ、各色差成分に対して5の、17値を、4:1:1サブサンプルされたデータに対して使

用でき、

-輝度に対して7且つ、色差成分に対して5の、12値を、4:1:1サブサンプルされたデータに対して使用でき、

-輝度に対して7且つ、各色差成分に対して6の、19値を、4:2:2サブサンプルされたデータに対して使用でき、そして、

-輝度に対して7且つ、色差成分に対して6の、13値を、4:2:2サブサンプルされたデータに対して使用できる。

【0134】 $21 \times 3 = 63$ ビットは8バイトより小さいので、量子化を転送し又は蓄積するのは少ない資源を使用する。中央処理ユニット（CPU）は、テーブルから1つの所定の量子化器を選択して、そして、それを、モーションJPEG 2000ビデオシーケンスの各フレームに関して特別目的のJPEG 2000ハードウェア（チップ）を制御する、CPU又は他のコントローラに書き込む。代わりに、JPEG 2000の1つの実行は、各フレームに対して選択されうる8又は16の異なる量子化器を保持する小さなメモリを有しても良い。

【0135】量子化器を、ビット面を階層に割当てるためにも使用できる。例えば、 Q_0 、 Q_1 及び、 Q_2 は、量子化する符号化パスのビット面を規定する量子化器でもよい。量子化器 Q_0 は、最も多くの損失を発生するが、一方量子化器 Q_2 は、最も少ない損失を発生する。階層1は、 Q_1 により量子化されたのではなく、全て Q_0 により量子化されたデータである。階層2は、 Q_2 により量子化されたのではなく、全て Q_1 により量子化されたデータである。階層3は、全て Q_2 により量子化されたデータである。

【0136】単純な量子化

図17と18は、いくつかの係数LSBを切り捨てるか又は符号化しない、3-レベル5/3変換に対する、例示の量子化器（ラベルA...Q）を示す。Nビット面の切捨ては、 2^N のスカラー量子化と等価である。前の量子化器に関して、量子化が変化するサブバンドは、破線の四角でハイライトされる。量子化器D、K及び、Qの全ては、サブバンド間の同じ関係を有する。MSEに関して又は他の歪基準に関してより良い、他の量子化器が使用され得る。

【0137】例示の以下のVerilog（ベリログ）は、単一の量子化値“q”を、7つの量子化器（切り捨てるLSBの数）に変換する。変数 q_1_HH は、レベル1HH係数に対して使用され、変数 q_1_H は、レベル1HL係数及びLH係数に対して使用される等である。幾つかのqの連続する値は、

【0138】

【外2】

$$i \geq 0$$

である全ての整数iに対して、同じ量子化器：0及び

1、2及び3、4及び5、8 i + 6及び8 i + 7とな * 【0139】
る。 * 【数8】

```
module makeQ(q, q_1HH, q_1H, q_2HH, q_2H, q_3HH, q_3H,
q_3LL);
    input [5:0] q;
    output [3:0] q_1HH;
    output [3:0] q_1H;
    output [3:0] q_2HH;
    output [2:0] q_2H;
    output [2:0] q_3HH;
    output [2:0] q_3H;
    output [2:0] q_3LL;

    wire [3:0] temp_2H;
    wire [3:0] temp_3HH;
    wire [3:0] temp_3H;
    wire [3:0] temp_3LL;
    wire [2:0] qlo;
    wire [2:0] qhi;

    assign qlo = q[2:0];
    assign qhi = q[5:3];

    assign q_1HH = qhi + ((qlo >= 2) ? 1 : 0);
    assign q_1H = qhi + ((qlo >= 4) ? 1 : 0);
    assign q_2HH = qhi + ((qlo >= 6) ? 1 : 0);
    assign temp_2H = qhi + ((qlo >= 1) ? 0 : -1);
    assign temp_3HH = qhi + ((qlo >= 3) ? 0 : -1);
    assign temp_3H = qhi + ((qlo >= 5) ? 0 : -1);
    assign temp_3LL = qhi - 1;

    assign q_2H = (temp_2H < 0) ? 0 : temp_2H;
    assign q_3HH = (temp_3HH < 0) ? 0 : temp_3HH;
    assign q_3H = (temp_3H < 0) ? 0 : temp_3H;
    assign q_3LL = (temp_3LL < 0) ? 0 : temp_3LL;

endmodule
```

色と周波数に対する人間の視覚システムの重み付け ※ニタ上の画像を視聴するのに適切であろう。より長い視
表9は、(J P E G 2 0 0 0規格のテーブルJ-2から 30 聴距離が、印刷画像又はテレビジョンに対して適切であ
の)人間の視覚システムの周波数応答を利用する輝度に ろう。
対して、量子化(例えば、切り捨て)する追加のビット 【0140】
面を示す。1000画素の視聴距離は、コンピュータモ ※ 【表9】

表9-輝度に対する人間の視覚システムの重み付け

サブバンド	視聴距離に対する量子化する余分なビット面			...
	1000 画素	2000 画素	4000 画素	
1HH	2	4 or 5	全て捨てる	
1HL, 1LH	1	2 or 3	6	
2HH	—	2	4 or 5	
2HL, 2LH	—	1	2 or 3	
3HH	—	—	2	
3HL, 3LH	—	—	1	

更に、色差は、輝度よりも更に重く量子化され得る。
【0141】図19は、図17(D)で開始し、且つ
(輝度と色差の両方に対して)1000画素子長距離に
対して周波数重みを加え、3LL色差が変更されないよ
うに保持し、4:2:2に関する1HLと1HH色差を
捨て、そして、残りの色差に対して追加の2ビット面が
捨てられる、量子化を示す。

【0142】リングングひずみの無いシャープなテキス
トは、テキスト/背景に対する正確な階調値よりも望ま
しい。即ち階調レベルが50%であり(例えば)、且つ 50

60%で挿入されているとすると、画像がテキストの場
合には、視覚的にしばしば好ましくないということはない。
一実施例では、LL(DC)係数は、低ビットレ
ートで、非テキスト画像よりも、テキストに対して更に重
く量子化される。例えば、8ビットの画像成分に対して
は、8、16又は、32の量子化ステップサイズがテキ
ストのみの領域に対して使用され、そして、1、2又
は、4の量子化ステップサイズがテキストを含まない領
域に対して使用されてもよい。これは、高周波数係数に
対する更に高い忠実度を可能とし、それにより、シャ-

ブなエッジを有するテキストとなる。

【0143】階層に分割するための量子化器の使用

表10は、16の例示の量子化器を示す。量子化器15

は無損失である。量子化器8は図19に示すものと同じ*

*である。これらを、サブバンドビット面を階層に分割するのに使用できる。

【0144】

【表10】

表10

サブバンド	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Y1HH	全て	全て	6	6	5	5	4	4	3	3	2	2	1	1	0	0
Y1HL,LH	6	5	5	4	4	3	3	2	2	1	1	0	0	0	0	0
Y2HH	5	4	4	3	3	2	2	1	1	0	0	0	0	0	0	0
Y2HL,LH	4	4	3	3	2	2	1	1	0	0	0	0	0	0	0	0
Y3HH	4	4	3	3	2	2	1	1	0	0	0	0	0	0	0	0
Y3HL,LH	4	3	3	2	2	1	1	0	0	0	0	0	0	0	0	0
Y3LL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C1HL,HH	4:1:1又は4:2:2に対してのみHLとHHは常に捨てられる															
C1LH	全て	全て	全て	全て	6	6	5	5	4	4	3	3	2	2	1	0
C2HH	全て	6	6	5	5	4	4	3	3	2	2	1	1	0	0	0
C2HL,LH	全て	6	5	5	4	4	3	3	2	2	1	1	0	0	0	0
C3HH	全て	6	5	5	4	4	3	3	2	2	1	1	0	0	0	0
C3HL,LH	全て	5	5	4	4	3	3	2	2	1	1	0	0	0	0	0
C3LL	全て	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C1HL,HH	4:1:1又は4:2:2に対してのみHLとHHは常に捨てられる															
C1LH	全て	全て	全て	全て	6	6	5	5	4	4	3	3	2	2	1	0
C2HH	全て	6	6	5	5	4	4	3	3	2	2	1	1	0	0	0
C2HL,LH	全て	6	5	5	4	4	3	3	2	2	1	1	0	0	0	0
C3HH	全て	6	5	5	4	4	3	3	2	2	1	1	0	0	0	0
C3HL,LH	全て	5	5	4	4	3	3	2	2	1	1	0	0	0	0	0
C3LL	全て	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

階層0は、量子化器0により離れて量子化されない全てのデータを含む。これは、3LLの全て、2HL、2LH、3HL、3LHの4ビット面と3HHを除き、2HHの5ビット面を除き、そして、1HLと1LHの6つのビット面を除く、輝度データのみである。階層1は、階層0内でなく、量子化器1により離れた量子化されていない全てのデータを含む。これは、色差3HLと3LHに対する5ビット面を除き；そして、色差2HL、2LH及び3HHに対する6ビット面を除く、1HLと1LHに対する輝度ビット面5、2HHに対するビット面4、3HLと3LHに対するビット面3；全ての3LL色差であろう。最後に、階層15は、1LH色差のLSBを含む。

【0145】複数の階層とタイル部分を伴うレート制御

圧縮システムのレート制御に関する幾つかの既知の技術がある。単純な方法は、圧縮された“ユニット”毎に歪を選ぶことである（ユニットはJPEGの8x8ブロック、動画シーケンス内のフレーム、単一画像のタイル、ウェーブレット符号化された画像内のタイルのサブバンド等である）。選択された歪が、望ましい平均ビットレートよりも高いビットレートを導く場合には、許される歪は、圧縮されている新たなユニットに対して増加される。選択された歪が、望ましい平均ビットレートよりも低いビットレートを導く場合には、許される歪は、圧縮されている新たなユニットに対して減少される。

【0146】更に複雑な方法は、幾つかの“ユニット”

からの圧縮されたデータをバッファする。各歪レベルでの各ユニットに対する、ビットレート及び／又は歪は、記憶される。そして、バッファ内の全てのユニットをわたって許される歪は、バッファが全て充填されているときに決定される。バッファが、全体の画像を含むのに十分である場合には、非常に高い品質を得ることができる。JPEG2000では、階層は品質への増加量を含めるように設計される。従って、歪を選択することは、各コードブロック又はタイルに対して使用する階層の数を選択することを意味する。この種のレート制御の完全な説明は、David Taubmanの画像処理に関するIEEE論文“EBCOTを伴う高性能スケーラブル画像圧縮”内に記載されている。

【0147】この処理には、幾つかの欠点がある。1つの欠点は、全体のコードストリームに対するバッファメモリが必要なことである。第2の欠点は、レーテンシー（待ち時、コードストリームが出力されるまでの時間）が高いことである。第3の欠点は、第2のパスが、大量の時間を使うことである。

【0148】これらの問題を緩和するために、JPEG2000コードストリームの各タイルは、少なくとも2階層で上述のように符号化される。各タイルの符号化の完了で、パケットの数（例えば、階層、分解能、区画、タイル成分）は、完全なタイル部分として、コードストリームへ出力される。残りの階層は、バッファ内に記憶される。バッファ内の残りの符号化されたデータを通した第2のパスは、オプションである。この第2のパスの

間に、各タイルからの余分なパケットは、空間又は時間
が許すなら完全なタイル部分としてコードストリームに
追加される。固定レートの実用の場合には、所定のレ
ート内のパケットのみが追加される。固定時間の応用で
は、唯一の数のサイクルが許される。この処理の一実施
例は、図15(A)に示されている。従って、これら
は、各タイルに対して2つの完全なタイル部分出力であ
る。

【0149】図15(B)は、幾つかの階層、階層1-
nを示す。階層1は、第1のパスの出力であり、そし
て、残りの階層は、固定時間又は固定レート時間制限以
下の可能性が高い。階層2は、全ての成分にわたって同
様な歪を達成しながら、固定時間又は固定レート要求内
の、第2のパスでの出力でもよい。

【0150】上述の処理は、バッファが符号化されたデ
ータの一部を記憶することを可能とし、第1のデータは
すぐに出力され(伝送され又は記憶され)そして、デー
タを通した第2のパスは処理するデータが少ないので高
速であるという、優位点がある。また、バッファは、少
ないメモリですむ。

【0151】どのパケットがタイル部分の第1の組みへ
行くかを選択する基準は、他のレート制御アルゴリズム
と同様である。一実施例では、パケットのレートは、全
体の画像の望ましい平均レートよりも小さい。例えば、
2.0bppでの最終的な圧縮ビットレートが望ましい
場合には、第1のパスはコードストリーム内の各タイル
に対して1.5bppを配置し、そして、各タイルに対
してバッファを1bppを配置する。

【0152】第2のパスは、残りのデータから各タイル
の第2のタイル部分内に配置するパケットを選択する。
従って、2.0bpp平均符号化を得るために、第1の
パスの後の高歪を有する幾つかのタイルは、そのタイル
に対して蓄積された残りのデータの全てを受信し、一
方、第1のパスの後の低歪を有する他のタイル部分は、
伝送される追加のデータを有しない。

【0153】圧縮されたコードストリームデータに対す
るレート制御

ここで説明する幾つかのレート制御技術は、コードスト
リーム内で保持する幾つかの数の階層を選択することに
より実行される要求に基づいて、圧縮されたコードスト
リームに行われるレート制御を含む。パーサは、階層
に基づいてビットレートを示す新たなコードストリーム

を発生するのに使用され得る。このビットレートは、要
求により規定されたビットレートと等しいか又はそれよ
り小さい。

【0154】パーサはここで、“パケット構造”と呼
ぶデータ構造を使用し得る。このデータ構造は、例え
ば、以下に示す多用途パケットデータ構造のような他の
目的に使用され得る注意する。一実施例では、このパ
ケット構造は、パケット開始ポイントとパケット長を含
む。また、タイル番号、解像度、成分、階層及びパケッ
トが属する区画も含む。最後に、選択フラグも有する。
このフラグは、所定の値(例えば、1)に設定されたとき
に、新たなコードストリームへ書き出す配列内で選択
されたかどうかを示す。

【0155】一実施例では、パケットはCODマーカー
により示された進行順序情報に基づいて、コードストリ
ームから連続する順序で読み出される。

【0156】このバイト数は、要求により望まれるビッ
トレートに基づいて計算される。階層0に属するバイト
の数は、合計に加算される。そして、この合計のバイト
数は、望ましいバイトの数と比較される。合計が望まし
いバイトの数よりも小さい場合には、1つの追加の階層
が全体に追加される。この処理は、合計が望ましいバイ
トの数と等しいか又はそれより大きくなるか又は、全て
のパケットは追加されるまで継続する。

【0157】この処理の間に、合計に追加されたそれら
のパケットは、構造内で選択フラグにより、選択された
としてマークされる。

【0158】合計が、望ましいバイトの数と等しい場合
には、追加の処理が停止される。合計が、望ましいバイ
トの数を超える場合には、追加された最後の階層内のパ
ケットは、合計から削除される。これは、ビットレート
が、望ましいビットレートよりも低いことを保証するた
めに、行われる。従って、減算ステップ中に、合計から
引かれたパケットは、未選択としてマークされる。

【0159】一実施例では、SOT、COD、PLTの
ような関連するマーカーは、要求に従って、更新され
る。パケットは新たなコードストリームに書き込まれ
る。パケット構造は、以下を使用して形成されても良
い。

【0160】

【数9】

```

39
typedef struct _PACK_ {          /* packet structure */
    int      start; /* packet starting point */
    int      length; /* packet length */
    unsigned short t; /* tile number the packet belongs to */
    unsigned short r; /* resolution the packet belongs to */
    unsigned short c; /* component the packet belongs to */
    unsigned short l; /* layer the packet belongs to */
    unsigned short p; /* precinct the packet belongs to */
    unsigned char  select; /* selection flag */
} Pack_t;

/* Store packets from tp->tile[i].Size[j] array to the packet structure array
**/* Layer progression (LRCP) order */

if(progression_order == 0){

    j = 0;
    for(i=0;i<number_of_tile;i++){
        m = 0;
        for(l=0;l<layer;l++){
            for(r=0;r<resolution+1;r++){
                for(c=0;c<component;c++){
                    for(p=0;p<precinct[r];p++){

                        tp->pk[j].start = tp->tile[i].pointer[m];
                        tp->pk[j].length = tp->tile[i].Size[m];
                        total_length += tp->tile[i].Size[m];

                        tp->pk[j].t = i;
                        tp->pk[j].r = r;

                        tp->pk[j].l = l;
                        tp->pk[j].c = c;
                        tp->pk[j].p = p;
                        m++;
                        j++;
                    }
                }
            }
        }
        num_packet[i] = m;
    }
}

```

多目的パケットデータ構造

一旦パケットが構造内に読み込まれると、上述したのと同じデータ構造を、他の解析オプションを容易にするのに使用することが可能である。

【0161】解像度解析では、除外されるべきパケットが、未選択としてマークされる。例えば、4解像度コードストリームが与えられ、そして、要求が3解像度コードストリームを生成することであるならば、パーサーは、解像度4に属する全てのパケットを、未選択として

マークする。そして、新たに発生されたコードストリームは、解像度1から解像度3までのパケットのみを含む。

【0162】同様に、成分解析、進行変換解析に対しては、品質解析を、その構造内でパケットを処理するステップごとに実行することが可能である。

【0163】パケットデータ構造は、複雑な要求を扱うことが可能である。例えば、要求は、パーサーに、3解像度、2階層そして、1成分コードストリームを

有するコードストリームを発生することを要求する。

【0164】各々の逆変換後のクリッピングウェーブレット係数に対して行われる量子化の結果、しばしば、最終的に符号化された画素は、規定のビット深さから、許された画素のものと範囲の外側にある。典型的には、これらの画素は、元の範囲にクリップされ、それにより、更なる画像処理又は表示装置は、元のビット深さを使用できる。

【0165】例えば、8ビット画像は、それを含めて0と255の間の画素値を有する。損失のある圧縮が使用された後に、復号された画像は、-5と256のような値を含み得る。8ビット出力を供給するために、これらの値は、それぞれ0と255にクリップされる。オリジナル画像が、クリッピング境界の外側の画素を含んでいなかったの、このクリップ手順は、通常は、画素に関する歪を減少する。この手順は、既知であり、JPEG 2000規格により推奨されている。

【0166】最終出力サンプルに関する境界に加えて、ウェーブレット変換の種々の段階で係数を仮定できる、値の境界がある。元の境界の外側にあるようにするために、量子化が、最終的に復号されたサンプルを変更することができるように、量子化は、元の球界の外側にあるように、変換されたウェーブレット係数を部分的に変換できる。それらの係数が、それらの元の境界にクリップされる場合には、歪は減少する。

【0167】例えば、8ビット入力サンプルを用いる、JPEG 2000により規定されている水平（1次元）5-3可逆変換後に、ローパス係数の最大値は、+191であり、そして、最小な可能な値は、-191である。ハイパス係数は、-255と255を含むその間でなければならない。垂直1次元変換後に、ローロー（Low-Low）係数は、-286から287である。従って、8ビット画像を復号するときに、第1レベルローローパス係数が（より高いレベルからの逆ウェーブレット変換により）発生され、係数は-286と+287にクリップされ、歪が減少される。同様に、第1レベル垂直逆変換が行われた後に、ローパス係数は、-191、+191にクリップされ、そして、ハイパス係数は、-255、255にクリップされる。

【0168】各サブバンド、各フィルタ、各変換レベル及び、各画像深さに対して、係数に対する最大値と最小値は異なっている。これらの最大値と最小値は、最大と最小を導く信号を見つけ、そして、前向き圧縮システムを実行し且つ最大値を記録することにより計算できる。極値を導くこの信号は、各画素が最大又は最小のいずれかである入力から来る。どの画素が最大であるべきで、且つどの画素が最小であるべきかは、ウェーブレット係数が負のときには、-1であるシーケンスを畳み込むことによりそして、ウェーブレット係数が負のときには、+1であるシーケンスを畳み込むことにより決定でき

る。JPEG 2000パートIで使用される5-3フィルタに対しては、対象のローパス信号は、 $[-1+1+1+1-1]$ であり、ハイパス信号は、 $[-1+1-1]$ である。

【0169】最も大きいLL値を発生する信号（画像）は、

【0170】

【数10】

+1 -1 -1 -1 +1
-1 +1 +1 +1 -1
-1 +1 +1 +1 -1
-1 +1 +1 +1 -1
+1 -1 -1 -1 +1

であり、ここで、+1は、入力最大値（例えば、255）により置換されねばならず、そして、-1は、入力最大値（例えば、0）により置換されねばならない。

【0171】不可逆フィルタに対しては、最大値を決定するためにシステムを実際に行う必要は無く、単にウェーブレット係数を畳み込むことで十分である。可逆5-3フィルタに対しては、しかしながら、係数の計算でfloor関数が使用され、そして、正しい最大値を決定するのに使用される。

【0172】これは、他のフィルタ（例えば、9-7フィルタ）に対して使用され得ることに注意する。

【0173】図28は、部分的に変換された係数に、クリッピングを伴う逆変換を適用するための、処理の実施例のフロー図である。この処理は、ハードウェア（例えば、回路、専用の論理、等）、（汎用コンピュータ又は、専用機械で実行される）ソフトウェア、又は、その両者の組合せを含む処理論理により行われる。

【0174】図28に戻ると、処理論理は、第1レベルの逆変換を、係数に行う（処理ブロック2801）。その後、処理論理は、部分的に変換された係数を所定の範囲にクリップする（処理ブロック2802）。次に、処理論理は、第1レベルの逆変換を、クリップされた係数に行い（処理ブロック2803）そして、部分的に変換された係数を、処理ブロック2802内の範囲と異なる所定の範囲にクリップする（処理ブロック2804）。再び、処理論理は、第1レベルの逆変換を、クリップされた係数に行い（処理ブロック2805）そして、部分的に変換された係数を、更に他の所定の範囲にクリップする（処理ブロック2806）。

【0175】単純化された色空間の取扱い

色管理を含む典型的な復号処理を図25に示す。図25を参照すると、制限されたICCプロファイルを含むファイルフォーマット（例えば、JPEG 2000規格内に記述されているファイルフォーマット）を有するファイルが、復号処理に供給される。逆圧縮ブロック2501は、ファイルのコードストリーム部分を取り、関係モデル化、エントロピー復号を実行し、そして、逆ウェー

ブレット変換を行うことにより、ファイルを逆圧縮するが、しかし、色変換動作を行わない。コードストリームがRCT又はICT成分を示す場合には、変換は、コードストリームを復号するのに使用されるべきであり、これらは、ブロック2502で実行される。即ち、逆RCT/ICTブロック2502は、成分と”RCTY/N”指示(yesならRCT、noならICT)を得て、そして、規定された逆変換を実行し、そして、(非表示の)RGB画素を供給する(構文で規定されている場合には、逆レベルシフトも実行される)。

【0176】最後に、表示装置に関する情報と共に、ファイルフォーマットからのICCカラープロファイルが、出力画素を発生するために使用される。

【0177】逆ICCブロック2503は、(非表示の)RGB画素とICCプロファイルを受信しそして、表示RGB画素に逆色変換を行う。

【0178】図26は、好適でないカメラ符号化器の一実施例を示す。図26を参照すると、カメラは、YCrCb画素を発生する。変換器2602は、YCrCb画素をRGB画素に変換し、そして、それらを、典型的なJPEG2000符号化器に供給する。符号化器は、RCT/ICT変換器2603を有し、圧縮器2604がそれに続く。圧縮器は、コードストリームに対するICC_Aを発生する。

【0179】図27は、単純なカメラ符号化器の一実施例を示す。即ち、RCT/ICT変換器2603と圧縮器2604を含む代わりに、単純なカメラ符号化器は、圧縮器ブロック2702のみを含む。図27を参照すると、カメラ2701は、YCrCb画素を発生し、そして、それらを圧縮器2702へ供給する。圧縮器は、RCT変換無し(JPEG2000符号化器を含み、そして、(逆RCTが復号に際して使用されるべきであることを示す構文を伴う)1と等しいRCTを伴うICC_B。コードストリームを発生する。ICC_BとICC_Aの間の関係は、以下の式、

【0180】

【数11】

$$ICC_B = ICC_A \cdot YCrCb^{-1} \cdot RCT$$

により与えられる。ここで、

【0181】

【外3】

は、関数合成を示す。

【0182】制限されたICCプロファイルは、画素の関数に対する”構文”である。カメラは、典型的には、全ての画素に対して同じプロファイルを書き込み、それにより、ICC_Bはオフラインで計算され、そして、各出力プロファイルにコピーされる。従来技術では、各画素に関して動作する、YCrCb⁻¹とRCT/ICTに関するHWがなければならない。

【0183】4:2:2:と4:1:1データを、量子化を伴う4:4:4データとして符号化

JPEG2000規格は、典型的には、4:4:4フォーマットのデータを扱うのに使用される。4:1:1又は、4:2:2フォーマットのデータを出力のために4:4:4フォーマットでどのように再構成するかを記述することは出来ない。一実施例では、4:1:1データを符号化するときには、符号化器は、1HL、1LH及び、1HH係数をゼロとして扱う。4:2:2データを符号化するときには、符号化器は、1HL、と1HH係数をゼロとして扱う。従って、余分なサブバンド内の全データは、ゼロに量子化され、復号器はその予想するようにコードストリームを受信できる。言いかえと、符号化されたデータは、重く量子化された4:4:4データに似ている。

【0184】サムネール、モニタ、プリンタ及び、全解像度及び、品質に対するファイル順序

複数の解像度での複数の画像は、多くの画像処理の状況において重要である。応用に依存し、ユーザは、異なる解像度の異なる画像選択することを望む場合がある。例えば、サムネール画像は、多数の画像のインデックスとして使用されうる。また、スクリーン解像度画像は、それを表示するモニタに送るのに使用され得る。プリント解像度画像はプリンタ応用に対する低解像度でもよい。

【0185】一実施例では、画像のコードストリームは、セクションに組織化され、それにより、例えば、サムネール版、スクリーン版、プリント版及び、無損失版のような画像の異なるバージョンは、品質により連続的である。

【0186】一実施例では、パケットは、あるパケットがサムネールのような特定の解像度に対応するように構成される。他のパケットとこれらのパケットの組合せは、モニタ解像度画像を表し、そして、他のパケットと組み合わせられたときにはプリンター版等を表す。POCとタイル部分を使用して、コードストリームの部分は、共にグループ化される。例えば、サムネールサイズの全てのタイルは、共にグループ化され、他の解像度のタイルが次に続き、そして他の解像度のタイルが次に続く等である。図21は、単一変換器対する、例示の進行を示す。各タイルのサムネールは、ファイルの先頭でタイル部分内にグループ化される。図21Aは、タイル部分2101のみが、サムネール画像に対して使用される部分であることを示す。図21Bは、モニタ解像度に対しては、タイル部分2102-2104が、タイル部分2101と共に含まれることを示す。図21Cは、プリンタ解像度に対しては、タイル部分2105と2106が、タイル部分2101-2104と共に含まれることを示す。最後に、図21Dは、データの無損失版には、残りの3つのタイル部分2107-2108が、他のタイル部分と共に含まれることを示す。タイル部分のこれらの

組みは、この進行順序でサーバ上に配置される。

【0187】タイル部分のグループ化されたものへアクセスする処理の一実施例を、図16に示す。その処理は、ハードウェア（例えば、専用の論理、回路等）、

（汎用コンピュータシステム又は、専用機械で実行される）ソフトウェア、又は、その両者の組合せを含む処理論理により行われる。以下のステップでは、画像は十分な解像度レベルと4つのサイズに画像を分割する階層で変換される。

【0188】図16を参照すると、処理論理は最初に、サムネールに対して正しい解像度と階層化を決定する（処理ブロック1601）。一実施例では、サムネールに対して正しい解像度と階層化を決定するために、処理論理は、各タイルに対してその解像度と階層に制約されるPOCを形成し、そして、タイル部分の組みを形成し、そして、コードストリーム内の各タイルに対してこのPOCを配置する。

【0189】次に、処理論理は、サムネールパッケージが既にコードストリーム内にあるとして、モニタ解像度に対して処理ブロック1601を繰り返す（処理ブロック1602）。そして、処理論理は、モニタパッケージが既にコードストリーム内にあるとして、プリンタ解像度に対して処理ブロック1601を繰り返す（処理ブロック1603）。

【0190】最後に、各タイルに対して、解像度と階層の端にPOCマーカを形成する（処理論理1604）。一実施例では、解像度と階層の端にPOCマーカを形成するのは、タイル部分の4つの組みと無損失変換のための残りのタイル部分を形成することにより実行される。

【0191】POC内に定義されるパッケージの特定の順序は、重要では無く単なる制限であることに注意する。

【0192】例示のコンピュータシステム

図20は例示のコンピュータシステムのブロック図である。図20を参照すると、コンピュータシステム2000は、例示のクライアント150又はサーバ100コンピュータシステムを有してもよい。コンピュータシステム2000は、情報を通信するための、通信機構又はバス2011と、情報を処理するためにバス2011に接続された、プロセッサ2012を有する。プロセッサ2012は、マイクロプロセッサに制限はされないが、例えば、PentiumTM、PowerPCTM、AlphaTM等のような、マイクロプロセッサを有する。

【0193】システム2000は、更に、プロセッサ2012をにより実行される情報と命令を記憶するバス2011に接続された、ランダムアクセスメモリ（RAM）又は、他のダイナミックメモリ装置2004（主メモリと呼ぶ）を有する。主メモリ2004は、プロセッサ2012をにより命令を実行する間に、一時的な変数又は他の中間情報を記憶するのに使用されてもよい。

【0194】コンピュータシステム2000は、プロセッサ2012をに対するスタティック情報と命令を記憶するためにバス2011に接続された読出し専用メモリ（ROM）及び／又は他のスタティックメモリ装置2006、磁気ディスク又は、光システム及び、その対応するディスクドライブのような、データ記憶装置2007も有する。データ記憶装置2007は、情報と命令を記憶するバス2011に接続されている。

【0195】コンピュータシステム2000は、更に、コンピュータユーザに情報を表示するために、バス2011に接続された、陰極線管（CRT）又は液晶ディスプレイ（LCD）のような表示装置2021を有する。英数字及び他のキーを含む英数字入力装置2022も、プロセッサ2012をへの情報とコマンド選択を通信するために、バス2011に接続されている。更なるユーザ入力装置は、方向情報とコマンド選択をプロセッサ2012へ通信しそして、ディスプレイ2021上でカーソルの移動を制御する、マウス、トラックボール、トラックパッド、スタイラス又は、カーソル方向キーのような、カーソル制御2023である。

【0196】バス2011に接続された他の装置は、ハードコピー装置2024であり、これは、命令、データ又は、他の情報を、紙、フィルム又は、同様な形式の媒体に印刷するのに使用される。更に、スピーカ及び／又はマイクロフォンのようなサウンド記録及び再生装置も、オプションでコンピュータシステム2000とオーディオインターフェースするバス2011に接続されてもよい。バス2011に接続されている他の装置は、電話又は携帯装置と通信する、有線／無線通信機能2025である。

【0197】システム2000にどの又は全ての構成要素と関連するハードウェアは、本発明の内で使用され得ることに注意する。しかしながら、コンピュータシステムの他の構成は、幾つかの又は全ての装置を含み得ることは理解されよう。

【0198】本発明は多くの変形と改良があることは、前述の説明を読めば当業者には明らかとなるが、説明した特定の実施例は、本発明を制限するものと解してはならない。従って、詳細な種々の実施例の参照は、本発明に必須であると考えられる構成のみを列挙している本発明の請求の範囲を制限するものではない。

【0199】

【発明の効果】本発明は、JPEG2000内の選択項目を使用することによる、高速、低コスト、小メモリ及び／又は、特徴の豊富な実行を可能とする、圧縮と逆圧縮機構の改善をおこなうことができる。

【図面の簡単な説明】

【図1】JPEG2000規格復号機構のブロック図である。

【図2】メモリ内の画像の構成の一実施例を示す図であ

る。

【図3】前向き変換に対して係数がどのように記憶されるかを概念的に示す種々のレベルに対する変換メモリ構成を示す図である。

【図4】入力画像データと画像データの種々の分解レベルを前向き変換と逆変換中に記憶できる単一メモリの実施例を示す図である。

【図5】入力データを扱う処理の一実施例を示す図である。

【図6】(A)は進行順序変換パーサーを有するシステムを示す図であり、(B)は解像度進行コードストリームから品質進行コードストリームへ変換する進行変換器を示す図である。

【図7】(A)は1つの進行順序から他の進行順序へコードストリームを変換する複数の経路を示す図であり、(B)は1つの進行順序から他の進行順序へコードストリームを変換する単純化された変換経路の一実施例を示す図である。

【図8】進行順序変換を実行する処理の一実施例を示す図である。

【図9】サイドバンド情報に基づいてコードストリームの部分を選択する復号器を示す図である。

【図10】復号時に階層を使用する処理のフローを示す図である。

【図11】編集処理の一実施例フローを示す図である。

【図12】特定の値に量子化される値の範囲のベル型カーブを示す図である。

【図13】フリッカを減少させる処理の一実施例を示す図である。

【図14】フリッカを減少させる量子化を実行する符号化器(又は、その部分の一実施例を示す図である。

【図15】(A)はレート制御を実行する処理を示す図であり、(B)は第1と第2のパスを受けうる例示の幾つかの階層を示す図である。

【図16】タイル部分のグループにアクセスする処理の一実施例を示す図である。

【図17】3レベル5、3変換に対する1つの成分に対する量子化器を示す図である。

【図18】3レベル5、3変換に対する1つの成分に対する量子化器を示す図である。

【図19】HVS重み付けされた量子化の例を示す図である。

【図20】コンピュータシステムの一実施例のブロックを示す図である。

【図21】単一サーバに対する、部分を伴う例示の進行を示す図である。

【図22】5、3不可逆変換に対する階層化の例を示す図である。

【図23】変換が5レベルを有し且つデータは階層0-3に分割されている例を示す図である。

【図24】第1と第3フレームの値が第2のフレームの値を設定するのに使用される、フリッカが避けられ得る状況の一例を示す図である。

【図25】色の取扱いを含む従来技術の復号処理のブロック図である。

【図26】好ましくないカメラ符号化器の一実施例を示す図である。

【図27】より単純なカメラ符号化器の一実施例を示す図である。

【図28】部分的に変換された係数にクリップを行う逆変換を行うための処理の一実施例のフローを示す図である。

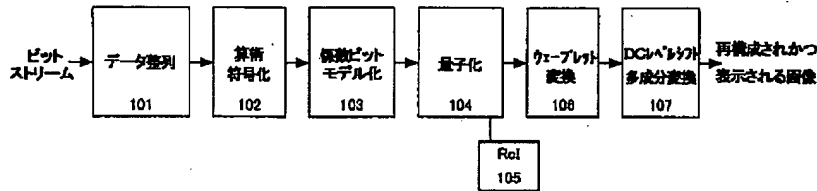
【符号の説明】

- 101 データ整列ブロック
- 102 算術符号化器
- 103 係数ビットモデル化ブロック
- 104 量子化ブロック
- 105 ROIブロック
- 106 変換ブロック
- 201 成分変換
- 202 メモリ
- 202 ウェーブレット変換処理論理
- 202A メモリアクセス論理
- 202B ウェーブレット変換
- 210 タイル
- 301 入力タイル
- 310、311 入力行
- 501 メモリ
- 502、502N 係数バッファ
- 503 関係モデル
- 504 エントロピー符号化器
- 505 符号化データメモリ
- 601 パーサー
- 602 サーバ
- 603 全画像
- 604 メモリ
- 901 復号器
- 902 量子化論理
- 903 メモリ
- 904 コードストリーム
- 905 復号論理
- 1400 量子化器
- 1401 量子化論理
- 1403 メモリ
- 1404 メモリ
- 1410 係数
- 2000 コンピュータシステム
- 2007 データ記憶装置
- 2011 通信機構又はバス
- 2012 プロセッサ

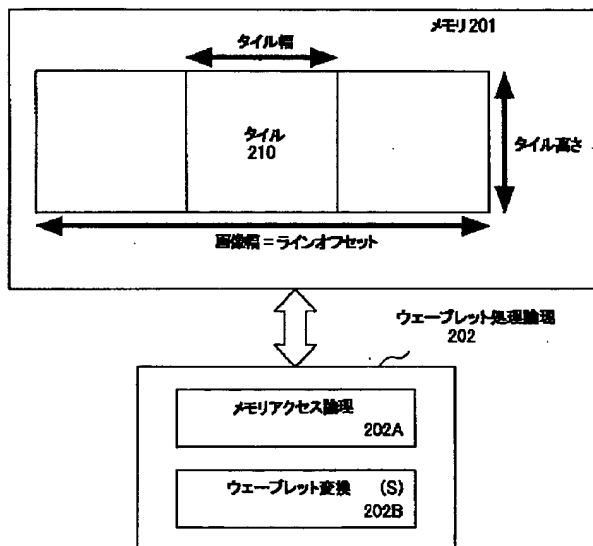
2021 ディスプレイ
2022 英数字入力装置
2023 カーソル制御
2024 ハードコピー装置
2025 有線/無線通信能力
2501 逆圧縮ブロック

* 2502 逆RCT/ICTブロック
2503 逆ICCブロック
2602 変換器
2603 RCT/ICT変換器
2604 圧縮器
* 2702 圧縮器ブロック

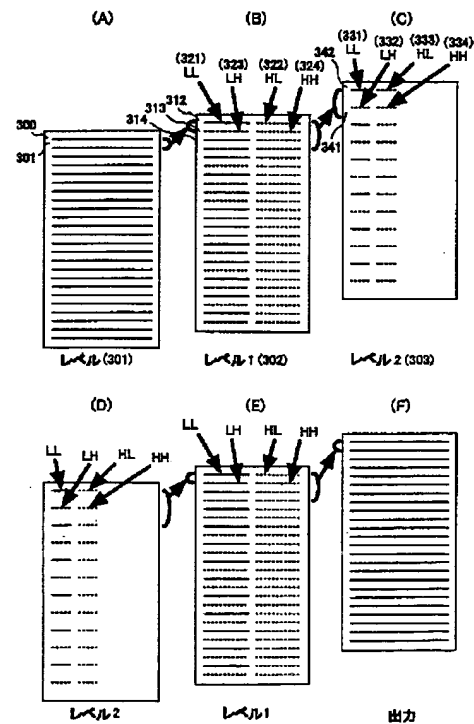
【図1】



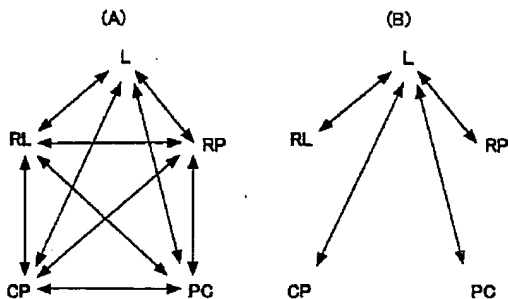
【図2】



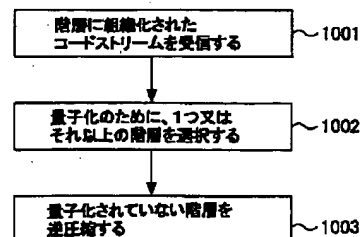
【図3】



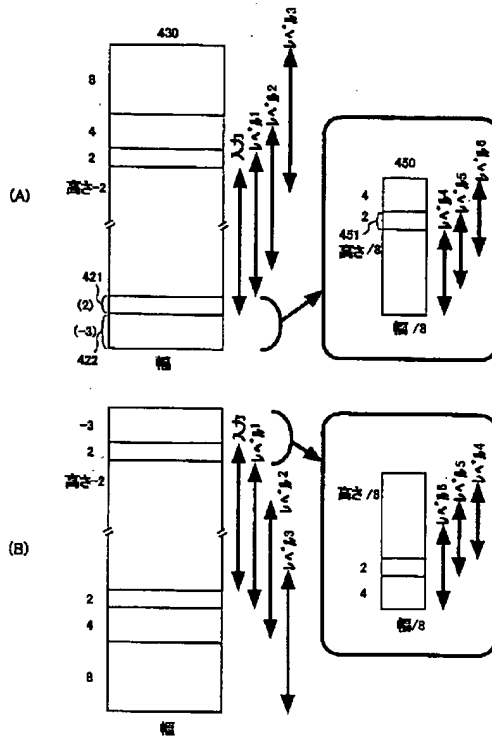
【図7】



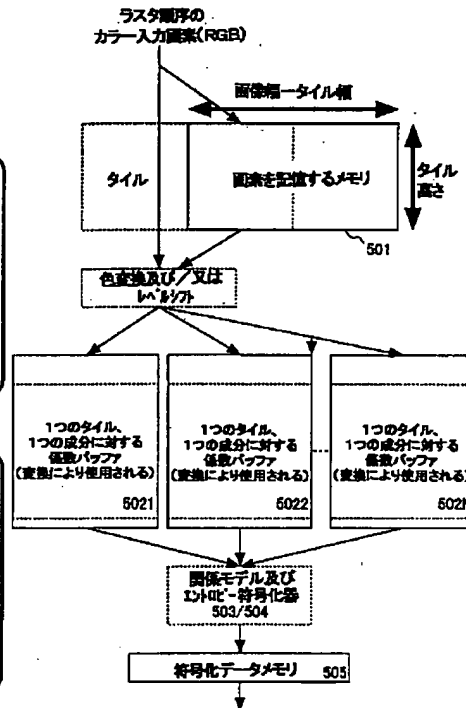
【図10】



【図4】



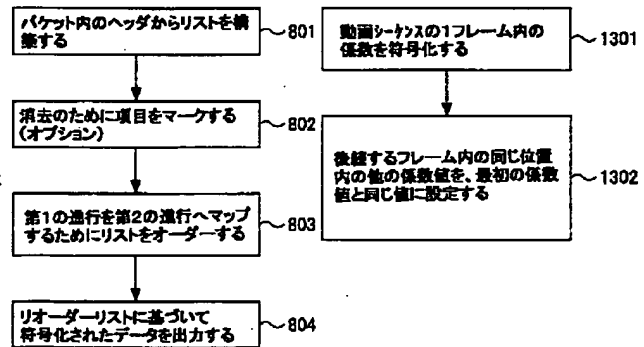
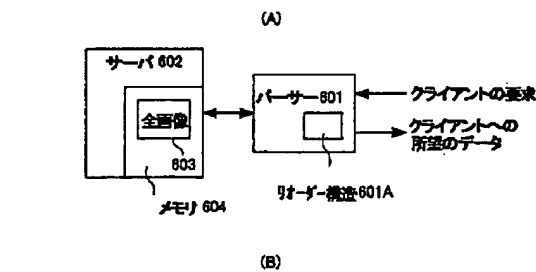
【図5】



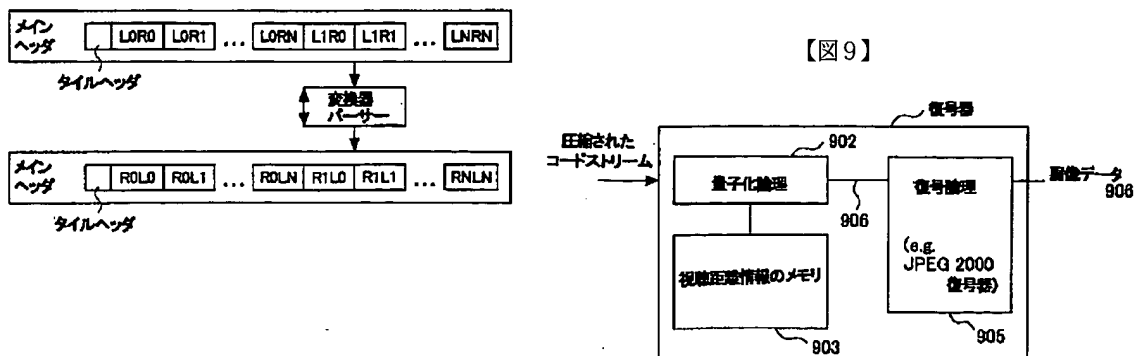
【図8】

【図13】

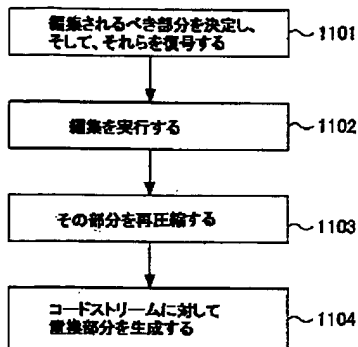
【図6】



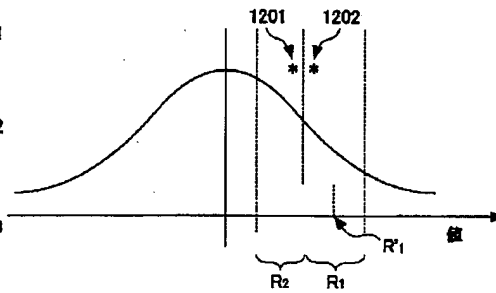
【図9】



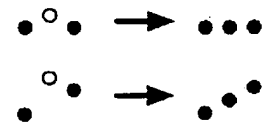
【図11】



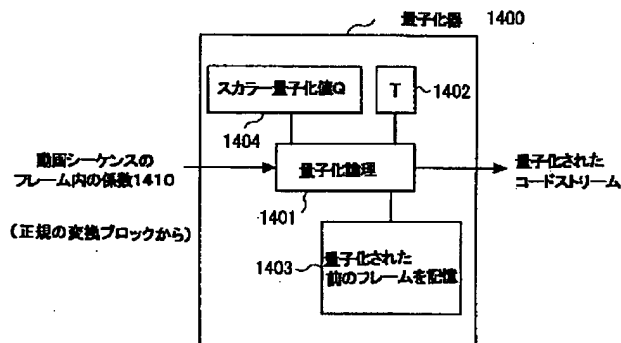
【図12】



【図24】

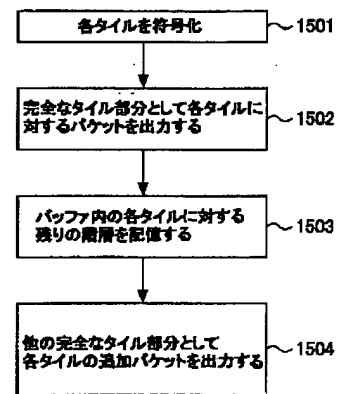


【図14】

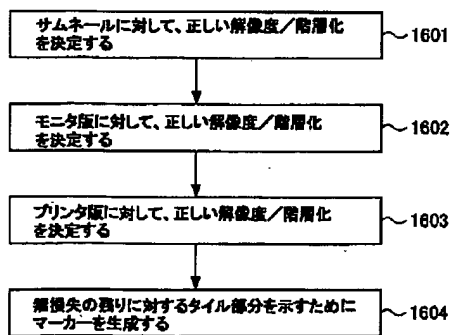


【図15】

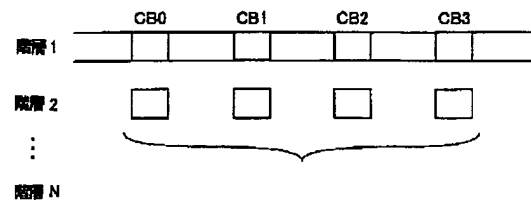
(A)



【図16】



(B)



【図19】

0 0	0		
0 0			2
0	1		
2	3		
2		3	

輝度

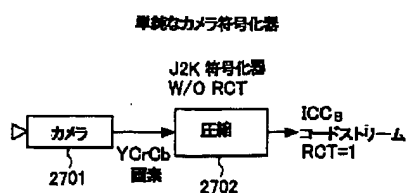
0 2	2		
2 2			all
2	3		
4			all
4			

色差

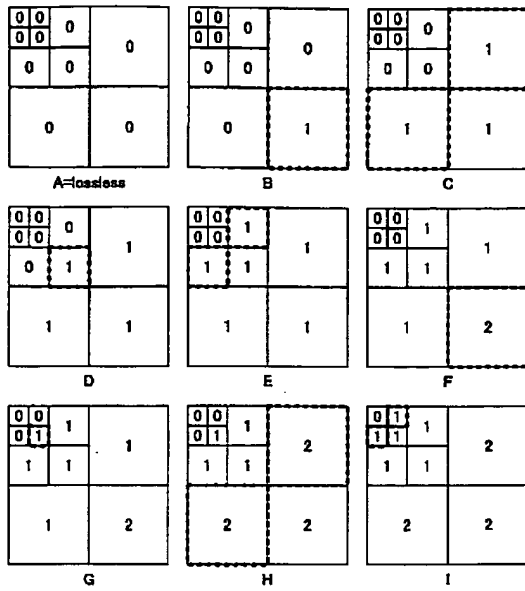
0 2	2		
2 2			all
2	3		
4			all
4			

色差

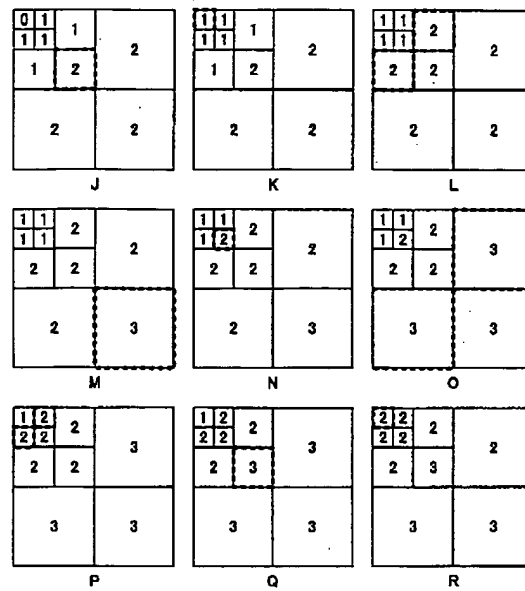
【図27】



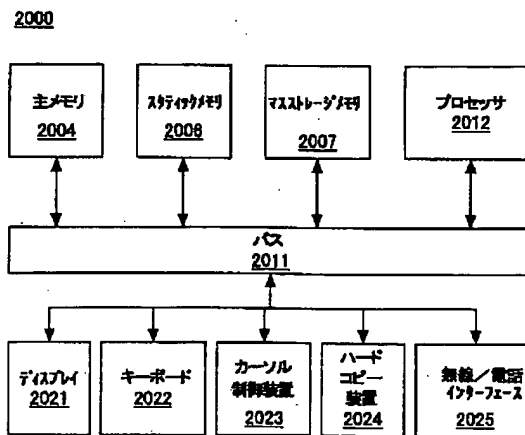
【図17】



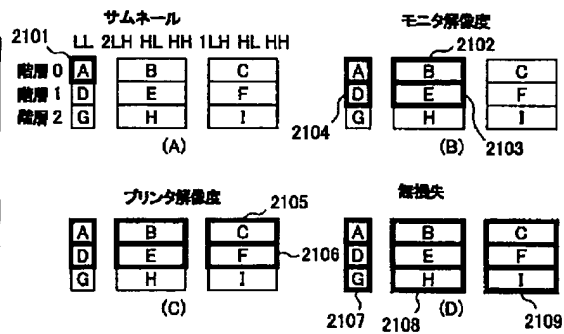
【図18】



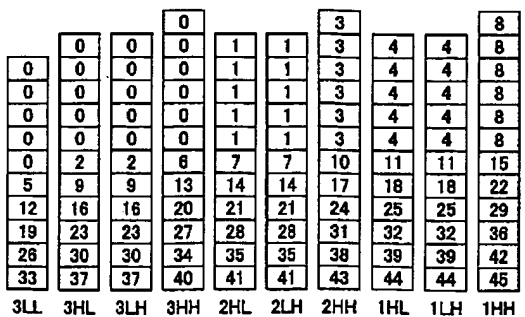
【図20】



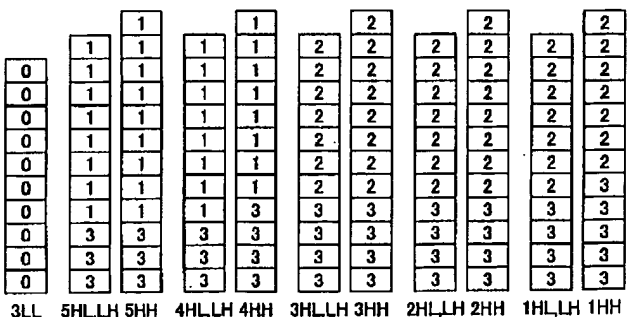
【図21】



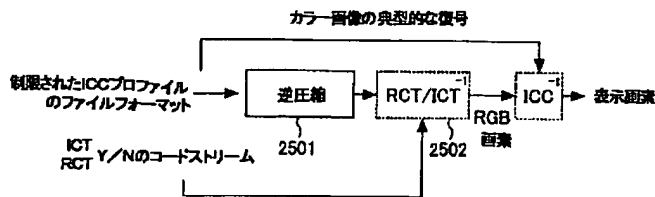
【図22】



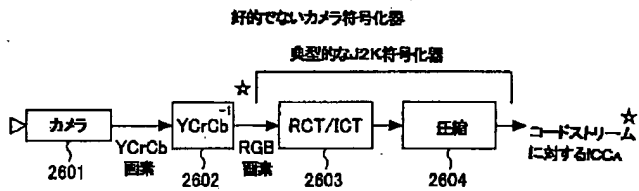
【図23】



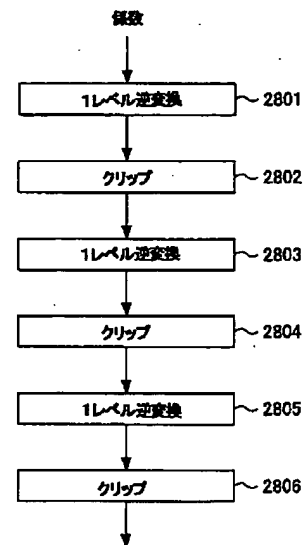
【図25】



【図26】



【図28】



フロントページの続き

(72)発明者 マイケル ゴーミッシュ
アメリカ合衆国、カリフォルニア州
94025、メンロ・パーク、サンド・ヒル・
ロード 2882番、スイート 115、リコー
イノベーション内

(72)発明者 マーティン ピー ボーリック
アメリカ合衆国、カリフォルニア州
94025、メンロ・パーク、サンド・ヒル・
ロード 2882番、スイート 115、リコー
イノベーション内

(72)発明者 ジーン ケー ウー
アメリカ合衆国、カリフォルニア州
94025、メンロ・パーク、サンド・ヒル・
ロード 2882番、スイート 115、リコー
コーポレーション内

Fターム(参考) 5B056 AA05 BB17 FF05 HH03
5C059 KK08 KK11 MA00 MA24 MC11
MC38 ME11 PP01 UA15 UA25
UA35 UA36 UA38 UA39
5C078 AA09 BA53 CA01 CA25 CA31
DA01 DA02
5J064 AA04 BA10 BA16 BB06 BC01
BC04 BC08 BC16 BD03

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-359850

(43)Date of publication of application : 13.12.2002

(51)Int.Cl. H04N 7/30

G06F 17/14

H03M 7/30

H03M 7/40

H04N 1/41

(21)Application number : 2002-035886 (71)Applicant : RICOH CO LTD

(22)Date of filing : 13.02.2002 (72)Inventor : SCHWARTZ EDWARD L
GORMISH MICHAEL
BOLIEK MARTIN P
WU GENE K

(30)Priority

Priority number : 2001 784928

Priority date : 15.02.2001

Priority country : US

(54) MEMORY USAGE SCHEME FOR PERFORMING WAVELET PROCESSING

(57)Abstract:

PROBLEM TO BE SOLVED: To improve a compression and inverse compression mechanism, capable of carrying out performance rich in characteristics with a high-speed, low-cost and small memory.

SOLUTION: A system comprising a memory and wavelet processing logic is described. The memory is sized to include a plurality of lines for storing a band of an image and additional lines. The wavelet processing logic comprises a wavelet transform and access logic. The wavelet transform generates coefficients, when it is applied to data in the memory. The access logic reads data from the memory into line buffers for supplying the data stored in the memory to the wavelet transform and to store coefficients in the memory, such that after the data stored at a first pair of lines are read from the memory, they are accumulated in the buffers of the access logic. The access logic reuses the first pair of lines to store coefficients generated by the wavelet transform, that are associated with a second pair of lines different from the first pair of lines.

LEGAL STATUS [Date of request for examination] 03.02.2005

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

*** NOTICES ***

**JPO and INPIT are not responsible for any
damages caused by the use of this translation.**

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The wavelet transform which generates a multiplier when it has the memory of size which includes Rhine which memorizes the band of an image, and two or more additional Rhine and is applied to the data in memory, To the line buffer supplied to wavelet transform, the data memorized in memory It has the access logic which reads data from memory and memorizes a multiplier to memory. After the data memorized in a pair of 1st Rhine are read from memory to the buffer of access logic, access logic The system which has the wavelet

processing logic which carries out the reuse of a pair of 1st Rhine which memorizes the multiplier generated by the wavelet transform relevant to a pair of 2nd different Rhine from a pair of 1st Rhine.

[Claim 2] Access logic is a system according to claim 1 which memorizes a multiplier in Rhine where memory continues with the multiplier from the decomposition level which is the same subband and adjoins mutually.

[Claim 3] Each 1st Rhine in a pair of 1st and 2nd Rhine is a system according to claim 1 which offsets mutually and is arranged in memory.

[Claim 4] Access logic is a system according to claim 3 which memorizes the 1st output of the wavelet transform to each multiplier level in Rhine of the addition within the distance of offset.

[Claim 5] The size of offset is a system according to claim 3 which is different on each conversion level.

[Claim 6] The size of offset is a system [equal to the conversion level of the multiplier memorized two times] according to claim 3.

[Claim 7] The offset for memorizing the 1st line of each set of the line of L1 multiplier in memory during decomposition The offset which is two lines from the 1st line of the data of the image relevant to said each set of the line of L1

multiplier, and memorizes the 1st line of each set of the line of L2 multiplier is a system according to claim 6 which is four lines from the 1st line of L1 multiplier relevant to said each set of the line of L2 multiplier.

[Claim 8] Access logic is a system according to claim 1 which memorizes the multiplier relevant to larger decomposition level than the level 3 in Rhine of the memory which memorized the band of an image before.

[Claim 9] Rhine of the addition relevant to offset is a system according to claim 3 which is above Rhine which memorizes the band of an image.

[Claim 10] Wavelet transform is a system according to claim 1 which is positive wavelet transform.

[Claim 11] Wavelet transform is a system according to claim 1 which is reverse wavelet transform.

[Claim 12] How to memorize the multiplier formed by applying wavelet transform to Rhine in memory so that each Rhine in memory might memorize from memory in a line buffer by the offset as which each class of the multiplier generated from read-out and the data memorized in Rhine of each set in memory requires data of Rhine of said each set in memory, in order to apply wavelet transform.

[Claim 13] After the data memorized in a pair of 1st Rhine are read into the buffer

of access logic from memory, in order to memorize the multiplier generated by wavelet transform It is the approach according to claim 12 which each 1st Rhine in a pair of 1st and 2nd Rhine is related mutually, offsets it, and is arranged in memory by having further the access logic relevant to a pair of 2nd different Rhine from a pair of 1st Rhine which carries out the reuse of a pair of 1st Rhine.

[Claim 14] The method according to claim 13 of having further the access logic which memorizes the 1st output of the wavelet transform to each multiplier level in Rhine of the addition within the distance of offset.

[Claim 15] The size of offset is a different approach according to claim 13 to each conversion level.

[Claim 16] The size of offset is an approach [equal to the conversion level of the multiplier memorized two times] according to claim 13.

[Claim 17] The offset for memorizing the 1st line of each set of the line of L1 multiplier in memory during decomposition The offset which is two lines from the 1st line of the data of the image relevant to said each set of the line of L1 multiplier, and memorizes the 1st line of each set of the line of L2 multiplier is the approach according to claim 16 of being four lines from the 1st line of L1 multiplier relevant to said each set of the line of L2 multiplier.

[Claim 18] The method according to claim 12 of having further the access logic which memorizes the multiplier relevant to larger decomposition level than the level 3 in Rhine of the memory which memorized the band of an image before.

[Claim 19] Rhine of the addition relevant to offset is the approach according to claim 13 of being above Rhine which memorizes the band of an image.

[Claim 20] When it is the manufacture which has the medium which accumulates the instruction which can be executed, and in which at least one record is possible and performs with a processor Each class of the multiplier generated from read-out and the data memorized in Rhine of each set in memory from memory data by the offset required of Rhine of said each set in memory in the line buffer which applies wavelet transform The manufacture which makes it perform to a processor to memorize the multiplier formed by applying wavelet transform to Rhine in memory so that it may memorize in each Rhine in memory.

[Claim 21] When it performs with a processor, in order to memorize the multiplier generated by it by wavelet transform after the data memorized in a pair of 1st Rhine were read into the buffer of access logic from memory by the processor It is the manufacture according to claim 20 which is made to perform carrying out the reuse of a pair of 1st Rhine relevant to a pair of 2nd different Rhine from a

pair of 1st Rhine, and has further the instruction which each 1st Rhine in a pair of 1st and 2nd Rhine is related mutually, offsets it, and is arranged in memory.

[Claim 22] Equipment with which each class of the multiplier generated from the means which reads data from memory in the line buffer which applies wavelet transform, and the data memorized in Rhine of each set in memory has ***** which memorizes the multiplier formed by applying wavelet transform to Rhine in memory by the offset required of Rhine of said each set in memory so that it may memorize in each Rhine in memory.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to use of memory for especially this invention to perform wavelet processing in relation to the field of compression and reverse compression.

[0002]

[Description of the Prior Art] New JPEG2000 decode specification (ITU-T Rec.T.800/ISO/IEC154441:2000 JPEG2000 image coding system) offers the definition of the new coding device and sign stream to an image. Although JPEG2000 specification is decode specification, by defining that a decoder must perform JPEG2000 prescribes coding and decode. Each image is divided into the tile (field) of the rectangle beyond one or it under JPEG2000 specification. When there are one or more tiles, tile-ization of an image generates the tile

component which carries out mutually-independent, and is extracted or is decoded. A tile component has all the samples of the predetermined component in a tile. An image has two or more components. Each of such a component has the sample of two-dimensional array. For example, a color picture has red, green, and a blue component.

[0003] A tile component is decomposed into different decomposition level using wavelet transform after tile-izing of an image. Such decomposition level contains some subbands occupied by the multiplier which describes the original horizontal and the perpendicular spatial frequency of a tile component. this multiplier -- the image whole -- crossing -- coming out -- there is nothing and the frequency information about a local field is offered. That is, a small number of multiplier describes a single sample completely. Decomposition level is connected with the following decomposition level with the space multiplier of 2 so that each continuous decomposition level of a subband may have the horizontal resolution of the abbreviation one half of front decomposition level, and half vertical definition.

[0004] Although there is a multiplier of a sample and the same number, the contents of information are in the inclination concentrated only on some

multipliers. The multiplier of many numbers decreases further through quantization. Processing of the addition by the entropy encoder is often large as compared with the original image, and decreases the number of bits required to express these quantized multipliers.

[0005] Each subband of a tile component is further divided into a code block. Grouping of these code blocks is carried out to a partition. These rectangle arrays of a multiplier are extracted independently. Entropy code modulation of each bit side of the multiplier within a code block is carried out with three coding pass. Each of these coding pass collects the information on the context about the bit side of the compressed image data.

[0006] Grouping of the bit stream of the compressed image data formed from these coding pass is carried out to a hierarchy. A hierarchy is the grouping of the arbitration of the coding pass which continues from a code block. Although there is big flexibility in hierarchization, a premise is that each continuing hierarchy contributes to the image of higher quality. The subband multiplier in each resolution level is divided into the rectangle field called a partition.

[0007] A packet is the fundamental unit of the compressed code stream. A packet contains the image data compressed from one hierarchy of one partition

of one resolution level of one tile component. These packets are arranged in the sequence defined within the code stream.

[0008] The code stream relevant to the tile systematized by the packet is arranged at one or the tile part beyond it. The tile partial header which consists of a series of markers and marker segments, or tags includes the information about various devices and coding formats required in order to decode [decide a location, extract and] and reconfigure for every tile. It becomes the head of all code streams from a marker and a marker segment, and there is a main header which also offers the information about an original image and the same information in it.

[0009] a code stream is boiled in the file format to which application makes it possible to translate the semantics of an image, and other information about it -- it can be contained. A file format may also contain other data of a code stream.

[0010]

[Problem(s) to be Solved by the Invention] A JPEG2000 code stream is performed by making sequence in a coding step reverse. Drawing 1 shows the block diagram of JPEG2000 specification coding device which acts about the compressed image data-code stream. If drawing 1 is referred to, a bit stream will

be first received by data alignment Block 101 who does re-grouping of the subband multiplier to a hierarchy. In order to decode the compressed bit stream, the related information collected during coding about the image data by which bit side compression was carried out, and its internal state are used for the algebraic-sign-ized machine 102.

[0011] After arithmetic decode, a multiplier is multiplier bit modeling Block 103, and receives bit modeling. Next, a code stream is quantized by quantization Block 104 quantized based on the target field (ROI) shown by ROI Block 105. After quantization, through conversion Block 106, inverse transformation is performed for the remaining multiplier, then component conversion Block 107 of DC and an option continues. Now, the reconfigured image is generated.

[0012] As for JPEG2000 specification, many selections remain for activation.

[0013]

[Means for Solving the Problem] A system including memory and wavelet processing logic is indicated. Memory has size including two or more Rhine, in order to accumulate the band (band) of an image, and additional Rhine. Wavelet processing logic includes wavelet transform and access logic. Wavelet transform generates a multiplier, when carried out to the data in memory. By supplying the

data memorized by memory to wavelet transform, access logic reads data from memory in a line buffer, in order to accumulate a multiplier into memory after the data memorized by a pair of 1st Rhine from memory to the buffer of access logic are read. Access logic carries out the reuse of a pair of 1st Rhine, in order to memorize the multiplier generated by the wavelet transform in relation to the 2nd different Rhine for Mino from a pair of 1st Rhine.

[0014]

[Embodiment of the Invention] I will be completely understood from the drawing of attachment of this invention, and the following detailed explanation. However, don't understand it as what a specific example is only for explanation and an understanding about this invention, and restricts that of this invention.

[0015] The improvement of compression and a reverse compressor style is indicated. In order to perform a high speed, low cost, small memory, and/or abundant activation of the description, it is the purpose of the technique indicated here and activation to use the selections in JPEG2000.

[0016] By the following explanation, in order to give perfect explanation of this invention, many details are given. However, this invention of perform [it / without these specific details] is clear to this contractor. For example, known structure

and equipment are shown by in the form of a block diagram instead of details, in order to avoid making this invention ambiguous.

[0017] Some parts of the following detailed explanation are shown by the algorithm of operation and notation expression about the data bit in a computer memory. These algorithm-publications and expressions are means used by this contractor of the field of a data-processing technique for transmitting the stereo of research to other contractors of these most efficiently. It is thought that an algorithm is the sequence of the step which has the self-consistency which draws the result of a request here. These steps need physical actuation of a physical amount. Usually, although it is unnecessary, these amounts are accumulated, transmitted, combined and measured and take the format of an electric or magnetic signal by which other actuation can be performed. From the reason of common use, these signals are mainly understood that it is often convenient to refer to as a bit, a value, an element, a symbol, the character, a term, a number, or same thing.

[0018] However, all these same vocabulary relates to suitable physical quantity, and should memorize that it is the convenient mere label given to these amounts. Unless it is said that it is especially clear from the following explanation, "it

calculates". ["it processing" or] Or "it determines". ["it computing" or] Or use of the vocabulary which lets "it displaying" or explanation like the same thing pass

The data shown as a physical (it is electric) amount in a computer system register and memory A computer system and memory, a register or other information storage, transmission Or being referred to at actuation and processing of the computer system or the same electronic calculation machine operated and changed into other data similarly shown as a physical amount in an indicating equipment is understood.

[0019] This invention relates also to the equipment which performs actuation explained here. This equipment may also contain the general purpose computer which is alternatively activated by the computer program which was constituted specially because of the required purpose, or was accumulated into the computer, or is reconfigured. Although the limit for which such a computer program and each were connected to the computer system bus is not carried out, the are recording medium in which computer read-out like a flexible disk, an optical disk, CD-ROM and a magnetic disk, read-only memory (ROM), random access memory (RAM), EPROM and EEPROM, the MAG, an optical card, or the medium of other formats suitable for memorizing an electronic instruction is

possible memorizes.

[0020] The algorithm and display which are shown here do not relate to a specific computer or other specific equipments at a proper. The equipment which performs the approach step as which various general-purpose systems can use it with a program according to the technique explained here, or are required and in which it specialized further can be realized conveniently. The required structure over these various systems will become clear from the following explanation. Furthermore, in addition, this invention is not indicated with reference to specific programming language. It will be understood that it can be used for various programming language performing the technique of this invention indicated here.

[0021] The medium in which read-out is possible includes by machine the device of the arbitration which memorizes the information on the format in which read-out is possible with a machine (for example, computer), and is transmitted. For example, the medium in which read-out is possible includes read-only memory ("ROM"), random access memory ("RAM"), a magnetic-disk are recording medium, an optical are recording medium, flash memory equipment, the propagation signals (for example, a subcarrier, an infrared signal, a digital

signal, etc.) of the format of electric, optical, acoustical, or others, etc. by machine.

[0022] The explanation below a general view relates to the activation or the new approach of using the flexibility of JPEG2000 or other coding devices of the same description.

[0023] Memory use drawing 2 about low memory and high-speed burst access shows one example of the configuration of the image in memory 201. When drawing 2 is referred to, only a "tile height" raster line has the band (band) of an image in memory 201, and a whole image does not have it. Thus, the amount of the image in memory 201 is equal to image width credit ***** height. Inside the band of an image, there is at least one tile like a tile 210.

[0024] The wavelet transform processing logic 202 has memory access logic 202A which reads data from memory 201 and memorizes data in memory 201 so that data may be given to wavelet transform 202B. The wavelet transform processing logic 202 has hardware, software, or the both combination.

[0025] In the one example, memory access logic 202A accesses a tile with four parameters of the Rhine offset (image width) for obtaining the head in other Rhine from the pointer to the head of the tile in memory or an index, the width of

face of a tile, the height of a tile, and one head in Rhine. Instead, memory access logic 202A can access a change of the width of face of a tile at memory 201 using the pointer or index to the last of a tile.

[0026] In the one example, in order to perform a certain function F and to access a part of each Rhine of a tile, or Rhine of an image, the following processings are used.

[0027]

[Equation 1]

1F of a function includes the thing of Rhine for which wavelet transform is given for constructing. DC level shift and multiplex component conversion are sufficient as other functions F.

[0028] Such processing is performed by hardware (for example, the logic of dedication, a circuit, etc.), software (it performs by the general purpose computer or the exclusive machine), or processing logic including the both combination.

[0029] In the one example, the multiplier from a subband is accessed using a

start point, width of face, height, and the same device in which it has the Rhine offset. Since both the lines of a multiplier are accumulated into memory, when it memorizes in a cache, burst accessible memory, or memory larger than one multiplier, a line is accessed efficiently.

[0030] Drawing 3 (A)-(C) shows the configuration of the conversion memory to various level which shows notionally how a multiplier is memorized. (The nomenclature of a 800/ISO/IEC15444 of ITU-T recommendations:2000JPEG2000 image coding system is used) All LH, HL(s), and HH multipliers are encoded. These multipliers are shown by the broken line within drawing 3 (B) and 3 (C). When having calculated conversion except for LL multiplier of the last conversion level encoded, only the input line and LL (shown by continuous line within drawing 3 [of continuous level] (B) and 3 (C)) multiplier of the input tile 301 need to be memorized temporarily. When the conversion which performs a horizontal and a perpendicular with one pass, and uses a line buffer is used, **** of an input row is read completely (an input row or LL multiplier), and the reuse of the space used by Rhine is once carried out.

[0031] Drawing 3 (A)-(C) shows the input tile 301, the level 1 (L1) (302) which aligned by offset, and level 2 (L2) (303) memory area, in order that how to face

how within one example may show whether it is attained. Addition of two lines of the lines 312 and 313 to the room used for holding the input tile 301 is needed for generating L1 multiplier, when carrying out the reuse of the memory to the input tile 301 to L1 multiplier. Addition of four lines of a line 341-342 to the room used for holding L1 multiplier is needed for generating L2 multiplier, when carrying out the reuse of the memory memory which memorizes L1 multiplier to L2 multiplier. (It is cautious of there being two lines of useless space among lines 341 and 342.) As for additional Rhine, it is desirable that wavelet transform is behind the direction given to the information in memory.

[0032] especially -- the input row of the input tile 301 -- constructing -- it corresponds to each one line of LL, LH, HL, and HH multiplier of level 1 of the result of having given the wavelet transform currently written in Rhine of the result of having given conversion to two different Rhine, and memory. For example, the result of having given wavelet transform to input rows 310 and 311 is a multiplier in the part of the lines 312 and 313 of L1 multiplier (302). For example, by the LL multiplier 321 of a line 312 being equivalent to LL multiplier (continuous line) of level 1, the HL multiplier 322 of a line 312 is equivalent to HL multiplier of level 1, and the LH multiplier 323 of a line 313 is equivalent to LH

multiplier of level 1, and the HH part 324 corresponds to HH multiplier of level 1.

The level 1 multiplier from the first two input lines is memorized in two excessive lines of the top of memory, and the remaining level 1 multiplier is written in the location which memorizes the data of the input tile 301, in order to carry out the reuse of the memory. The width of face and the height to each format (for example, LH, HL, HH) of the multiplier to each subband of a level 1 multiplier are the one half of the input tile 301. The Rhine offset (for example, distance of a line 312 to the line 314 in drawing 3 (B)) of LL line of LL line about level 1 to a degree is twice the tile width of face (since each multiplier line is from the field corresponding to two Rhine).

[0033] Similarly, it is a multiplier in two lines, i.e., the result (continuous line) of having given wavelet transform to two lines of LL multiplier in level 1 is LL multiplier (331), LH multiplier (332), HL multiplier (333), and HH multiplier (334) in level 2. The width of face and the height of level 2 multiplier are the quadrant of the input tile 301. The Rhine offset over level 2 is 4 times the tile width of face (since each multiplier line is from the field corresponding to two level 1 LL lines or four input lines). Thus, four excessive Rhine of memory is needed for using the same memory which memorizes an input tile, in order to memorize L2

multiplier. When 3rd decomposition level is performed, it is cautious of eight lines of an addition being the need. Therefore, in this example, in order to make it possible to carry out the reuse of the memory which has two level of the decomposition which memorizes input time amount and is given there, excessive Rhine of 14 is needed in total. determining excessive Rhine -- child merit **** -- a general formula is given by the following $2^{(\text{maxlevel}+1)}-2$.

[0034] In order to access a subband like LL, LH, HL, and HH subband, it needs to be offset between a head pointer, and a line/Rhine. Height and width of face are also required in order to know the time of stopping when accessing a tile.

[0035] If decomposition level increases, some lines in memory will become intact. That is, it is that Rhine of the memory under L1 multiplier after the 1st decomposition level becomes intact, and Rhine of the memory under L2 multiplier after the 2nd decomposition level becomes intact etc. The reuse of the excessive space is carried out in the one example.

[0036] Drawing 3 (D) -3 (F) memorizes the result which additional Rhine gave inverse transformation, and Rhine of these additions shows the corresponding inverse transformation memory use behind [that inverse transformation is performed within memory] a direction.

[0037] Drawing 4 (A) is drawing showing the example of the single memory which can memorize an input and various decomposition level to positive conversion, respectively. Reference of drawing 4 (A) shows the location to an input tile, a level 1 multiplier, level 2 multiplier, and level 3 multiplier to 2 of an addition, 4, and eight lines, respectively. Drawing 4 (B) is drawing which the input loading factor of the various level of conversion can memorize with an output during inverse transformation and in which showing the example of the same single memory.

[0038] Table 1 shows the amount of the memory demanded from the various conversion level of 256x256 tile to separate memory and the memory by which the reuse was carried out.

[0039]

[Table 1]

表 1

レベル	別のメモリ (バイト)		再使用されたメモリ (バイト)	
1	256x256 =	65,536	2x256 =	512
2	128x128 =	16,384	4x256 =	1,024
3	64x64 =	4,096	8x256 =	2,048
4	32x32 =	1,024	16x256 =	4,096
5	16x16 =	256	32x256 =	8,192
6	8x8 =	64	64x256 =	16,384

The amount listed to the memory by which the reuse was carried out is the new memory of the addition used to the level. Level 1 and the reuse of memory to 2 and 3 save memory to this example. Level 4 uses another memory.

[0040] the inside of the single memory after, as for the memory to level 4, 5, and 6, level 3 was generated -- or and it differs completely, it can arrange in another memory. The amount of required memory is 38×32 , and it is smaller than 5×256 .

Since there is two intact Rhine (namely, memory which memorized the last two Rhine of input data) after generating of a level 1 multiplier, saving of small memory can be attained by carrying out the reuse of these two Rhine on level 4, 5, and 6. The numbers of Rhine of the addition to level 4, 5, and 6 are 16, 32, and 64, and the excessive space between Rhine is separated from front level 2 times, and since it is half width of face, especially this is important.

[0041] In the one example, the multiplier from level 4, 5, and 6 is put in small memory structure like the storage region 450 of drawing 4 . Reference of drawing 4 memorizes level 4 multiplier in the field which has height equal to the tile height divided by 8 (3 level 23), and has width of face equal to the width of face w divided by 8 (it is 23 and 3 corresponds to the number of the level memorized before). Two additional Rhine 451 is need altogether memorizing the

multiplier of level 5 in the same required storage region. Similarly, it is the need altogether holding four lines of an addition to the multiplier of level 6 using this memory storage region. When memorizing a multiplier, it is cautious of not skipping Rhine. In the one example in which 256x256 tile is processed, with 421, 422 [of five lines / of two lines] of about 4.75 lines is used, although [which holds a storage region 450] it is excessive in the low section of the are recording field 430. As shown, in addition to the allocated memory or it, the approximation by 422 shows the memory of three lines required to memorize an input tile. The reuse of the field which memorizes an input tile by this approach is carried out nearly completely.

[0042] In the one example, very much, a few or in order to use the minimum memory potentially, level 6 is independently memorized from level 4 and 5. However, this only saves 64 bytes of memory.

[0043] Small memory can hold all the transform coefficients of 256x256 tile for a while rather than 273x256. This is smaller than a very suitable memory configuration 7%. Unlike a suitable memory configuration, the line which the excessive copy was avoided and was put in coincidence on the other hand for [both] rapid access is held.

[0044] Table 2 shows other examples of the another memory and the memory by which the reuse was carried out to 128x128 tile. To this size, the first three conversion level can carry out the reuse of the memory in 142x128 buffer.

[0045]

[Table 2]

表 2

レベル	別のメモリ (バイト)	再使用されたメモリ (バイト)
1	128x128 = 16,384	2x128 = 256
2	64x64 = 4,096	4x128 = 512
3	32x32 = 1,024	8x128 = 1024

The decision which uses suitable memory or new memory in the one example is the height of a tile, and the function of conversion level. Such decision is based on below. Tile height > In the case of 2 ($3 \times \text{level} - 2$), a suitable approach is used. Tile height = in the case of 2 ($3 \times \text{level} - 2$), either is used. In the case of tile height < 2 ($3 \times \text{level} - 2$), new memory is used.

[0046] Table 3 is as follows in order to show application of decision.

[0047]

[Table 3]

表3

レベル	$2^{(3 \times \text{level} - 2)}$
1	2
2	16
3	128
4	1024
5	8192

It is inconvenient to fit a memory configuration to tile height in some application.

The memory configuration of single immobilization can be used. Tile size smaller than 128x128 serves as compressive ability which is not desirable typically, therefore, typically, is not used. However, when bigger tile size than 1Kx1K is used to a very big image, this does not improve compression greatly but, typically, a lot of memory needed has a heavy burden. Therefore, the tile height between 128-1024 is assumed, and it is good to use suitable memory to 3 level of conversion in fact.

[0048] Decode is the same, before decode logic reads, the result of having given inverse transformation is written in and a remarkable difference is only that initiation is started from a record level from level 6 to level 1 in an above-mentioned example to the minimum level. In such a case, an input tile finishes with the top of memory structure. Excessive Rhine in which the reuse of memory is held is descending order. For example, eight lines is required to use

the structure of drawing 4 (B) and form L2 multiplier from L3 multiplier, four excessive Rhine is required to form L1 multiplier from L2 multiplier, and two excessive Rhine is required to form an input tile from L1 multiplier.

[0049] In the one example, in order to treat input tile data, color conversion may be performed in advance of coding. Drawing 5 shows one example of processing treating input data. Reference of drawing 5 receives a color input pixel in order of a raster. These color pixels are RGB, YCrCb, CMY and CMYK, gray scale, etc. A color input pixel may be memorized as a tile in memory like memory 501 with a band (or other formats).

[0050] The pixel received directly from an input from the phase 501 receives color conversion and/or a level shift, and the output of a result is memorized by one 5021 to 502 N multiplier buffer. That is, once color conversion is completed to each tile, one 5021 to 502 N multiplier buffer will memorize, and the following tile will be processed. In the one example, there is a multiplier buffer to each component.

[0051] A 5021 to 502 N multiplier buffer is used by conversion, as it mentioned above in order to perform wavelet transform, carrying out the reuse of the memory. Therefore, an input and the output of a 5021 to 502 N multiplier buffer

are wavelet transforms.

[0052] After conversion is given to a 5021 to 502 N multiplier buffer, a relational model 503 and the entropy encoder 504 perform further compression processing about the already changed data. The buffer of the encoded data is carried out into the coding data memory 505.

[0053] Conversion may be applied to another **** tile, performing further compression processing to one tile. Similarly, actuation of all which **** may be performed to coincidence at two or more tiles.

[0054] The data in the compressed code stream are memorized by one of the five advance sequence, and are sold at advance sequence conversion JPEG2000 specification. Advance sequence may change in that it differs in a code stream. Sequence is defined by embedded "for layers (hierarchy)" about a hierarchy, a partition, resolution, and a component.

[0055] Five advance sequence is indicated in the specification in the table A-16 of JPEG2000 specification. They are hierarchy-resolution-component-location advance (LRCP), resolution-hierarchy-component-location advance (RLCP), resolution-location-component-hierarchy advance (RPCL), location-component-resolution-hierarchy advance (PCRL), and

component-location-resolution-hierarchy advance (CPRL).

[0056] Sequence is defined within COD within JPEG2000 specification, or a POD marker. A coding format default (COD) marker is defined by JPEG2000 specification, and describes the hierarchization which is the default used for compressing a coding format, decomposition level and an image, or (if it being in a tile partial header) all the components of a tile (if it being in a main header). An advance sequence modification (POD) marker describes a boundary and advance sequence to every advance sequence except being specified in the COD marker segment in a code stream. A packet size main header (PLM) shows the location which shows the list of packet sizes in a tile part, and a packet size to each tile part in order, and a tile PERT header (PLT) shows the inside of a tile part, and a tile packet size, and has data in a code stream.

[0057] B. To predetermined advance sequence, how is constituted or, as for JPEG2000 specification in Chapter 12, the packet of compressed data specifies a chisel. It does not describe how data are changed into other advance sequence from one advance sequence.

[0058] In the one example, an advance sequence conversion parser changes a code stream in order of a desired advance based on a user's input, without

carrying out data decode and encoding it again. Drawing 6 (A) shows the system which has such a parser. If drawing 6 (A) is referred to, a parser 601 will receive the demand from the client about specific advance sequence. This client chooses a specific link, looking at a web page. Answering this demand, a parser 601 accesses a server 602 and changes in order of advance which is different in a code stream based on code stream **** relevant to all the images 603, and a demand from memory 604. This demand uses an options command (for example, RL2L (from resolution-hierarchy advance to hierarchy advance) advance sequence is shown.). The advance sequence described is based on a hierarchy, resolution, the component, the partition, or the tile.

[0059] Drawing 6 (B) shows the advance converter changed into a resolution advance (RLCP) code stream from a hierarchy advance code stream (LRCP). The map of the advance sequence is carried out directly mutually.

[0060] Drawing 7 (A) shows two or more paths of changing a code stream into other advance sequence from one advance sequence. If drawing 7 (A) is referred to, each of five advance sequence (LRCP, RLCP, RPCL, CPRL, and PCRL) is shown with the path to each other, and, thereby, all advance is shown. In the one example, a parser performs hierarchy advance to all conversion, and

performs through and selected conversion first. Drawing 7 (B) shows one example of such a simplified conversion path, and the number of mapping demanded decreases from 10 (it is shown in drawing 7 (A) like) to 4. However, among five advance sequence, before one of the throats reaches the selected sequence, it can be used as one from which all are changed. The conversion technique explained here simplifies a source code so that the number of Rhine of a source code may become less than conversion of many paths very much. This decreases debugging time amount and brings little memory and a result of an execution-time variable.

[0061] In order to perform conversion, the reorder of the sequence of the packet in a code stream is carried out, and if it is ****, there is nothing. As for a packet, a label is attached by the continuation sequence within a code stream. A marker shows [the start point of data, the die length (or instead of, ending point of data) of data, and] how data should be treated. for example, a ***** [whether as for those directions with which data should be treated how, data should be deleted, and that data should be omitted] -- or it is shown in data whether other actuation should be performed. Such handling information comes also from rate distorted information which is established in PLT/PLM of JPEG2000 specification, and/or

a PPT/PPM marker set. It is possible to omit without a code stream changing a packet header by this approach.

[0062] one example -- a list and an array -- or -- others (it is (like the reordering structure of 601A)) -- structure is built by showing the part of the data in each packet. This structure is used and the reorder of the packet is carried out.

[0063] Drawing 8 shows one example of the processing which performs advance sequence conversion. This processing is performed by hardware (for example, the logic of dedication, a circuit, etc.), software (it performs by the general purpose computer or the exclusive machine), or processing logic including those both combination.

[0064] If drawing 8 is referred to, it will start by the processing logic which builds the list from the header in a packet (processing Brock 801), and "processing will eliminate" the Mirkin grist item optionally for quantization (processing Brock 802). Next, the reorder of a list is performed in order that processing logic may carry out the map of the original advance to a desired (it includes treating an input and an output by advance to which it was specified by the POC marker (boundary about advance sequence)) advance (processing Brock 803). After that, processing logic outputs the data encoded based on the reorder list (processing

Brock 804).

[0065] Therefore, the combination of a reorder and analysis makes possible specifications, such as desired reordering and resolution, and quality.

[0066] Below the example of advance sequence conversion is an example which shows how a packet is arranged within a code stream. The code stream is formed based on two components, two hierarchies, 3 decomposition level, and hierarchy advance.

[0067] Table 4 shows packet sequence, die length, and the related index of the packet within this example. A packet sequence train shows the continuation sequence of the packet arranged in a code stream. Die length shows the die length of a packet. A related index shows the partition of resolution, a hierarchy, a component, and a packet.

[0068] For example, a packet [0] is a packet of the beginning in the code stream after the 1st tile header. It has 589 bytes. The related index $RwLxCyPz$ shows resolution w , Hierarchy x , Component y , and the packet adapted to Partition z .

[0069]

[Table 4]

表 4

パケット順序	長さ	関連インデックス
packet[0]	length=589	R0L0C0P0
packet[1]	length=589	R0L0C1P0
packet[2]	length=924	R1L0C0P0
packet[3]	length=924	R1L0C1P0
packet[4]	length=1602	R2L0C0P0
packet[5]	length=1602	R2L0C1P0
packet[6]	length=733	R3L0C0P0
packet[7]	length=733	R3L0C0P0
packet[8]	length=535	R0L1C0P0
packet[9]	length=535	R0L1C1P0
packet[10]	length=1523	R1L1C0P0
packet[11]	length=1523	R1L1C1P0
packet[12]	length=5422	R2L1C0P0
packet[13]	length=5422	R2L1C1P0
packet[14]	length=16468	R3L1C0P0
packet[15]	length=16468	R3L1C1P0

Within this code stream, grouping of the packet is carried out based on the hierarchy with them. The first eight packets belong to a hierarchy 0. Eight continuing packets belong to a hierarchy 1.

[0070] Transform processing explained here is used and an above-mentioned code stream is changed into resolution hierarchy advance. In the following, the packet above-mentioned [how] shows whether a reorder is carried out.

[0071] After a hierarchy advance code stream is changed into resolution advance, based on resolution, grouping of the packet is carried out within a new code stream. Such grouping is shown in Table 5. It is that the first four packets belong to resolution 0, and the four following packets belong to resolution 1 etc.

[0072]

[Table 5]

表 5

前のパケット順序	パケット順序	長さ	関連インデックス
0	packet[0]	length=589	R0L0C0P0
1	packet[1]	length=589	R0L0C1P0
8	packet[2]	length=535	R0L1C0P0
9	packet[3]	length=535	R0L1C1P0
2	packet[4]	length=924	R1L0C0P0
3	packet[5]	length=924	R1L0C1P0
10	packet[6]	length=1523	R1L1C0P0
11	packet[7]	length=1523	R1L1C1P0
4	packet[8]	length=1602	R2L0C0P0
5	packet[9]	length=1602	R2L0C1P0
12	packet[10]	length=5422	R2L1C0P0
13	packet[11]	length=5422	R2L1C1P0
6	packet[12]	length=733	R3L0C0P0
7	packet[13]	length=733	R3L0C1P0
14	packet[14]	length=16468	R3L1C0P0
15	packet[15]	length=16468	R3L1C1P0

It is [0073] to the hierarchy advance from the one example resolution of a conversion algorithm.

[Equation 2]

```

n = 0;
for(l=0;l<layer;l++){
    for(r=0;r<resolution+1;r++){
        for(c=0;c<component;c++){
            new_packet[n] = old_packet[l*component + r*layer*component +
c];
            n++;
        }
    }
}

```

Layer to Resolution Progression

```

n = 0;
for(r=0;r<resolution+1;r++){
    for(l=0;l<layer;l++){
        for(c=0;c<component;c++){
            new_packet[n] = old_packet[r*component +
l*(resolution+1)*component + c];
            n++;
        }
    }
}

```

Here, they are the number of the hierarchy in a layer= code stream, the number of the decomposition level in a resolution= code stream, and the number of the component in a component= code stream.

[0074] It is data concealment (side band information) to that of JPEG2000 coding.

Bit concealment makes it possible to transmit side band information, without making a file size increase. (The COM marker defined by JPEG2000 may be

used instead) Although a file size is increased, the side band information which does not intercept a simple decoder is also important.

[0075] Padding of some marker segments, packet headers, and packets is carried out to a cutting tool also with near origin. The examples of a JPEG2000 marker segment are PPM, PPT, PLM, and PLT. In addition, the twist required to include QCD, QCC, and POC of some marker segments may also be still longer. The padding data value is not defined in all these cases.

[0076] Although a limit is not carried out in the data of the undefined with which the coding device of some dedication has been arranged at this half-random, it can be used in order to offer the information including decode and filtering directions, ownership, segmentation directions, etc. on some important formats. Directions may also contain the index to a specific improvement device. For example, when the image is known as it is almost a text, the value which shows that the 1st pretreatment filter should be used is sent. On the other hand, when a field is mostly a graphic image, the value which shows that the 2nd pretreatment filter should be used is sent.

[0077] The following is the location where side band information may be accumulated, or a bit is hidden in a code stream.

- after the packet of the last of the packet header rounded off by the algebraic-sign-ized machine (AC) terminal point without a terminal point which can be predicted, and - cutting tool, and the - last -- before the following tile and - they are the tag tree structure by always not using min, a - packet header L Brock signal, the LSB parity (amelioration pass -- clean-up pass) to - code block, -QCD, a QCC excessive subband, and POC.

[0078] For example, the data which use AC terminal point carry out concealment Seki, and at least, whenever termination of the encoder is carried out, 0 to 7 bits is prepared. However, this is extended to some cutting tools. These special bits and cutting tools are used for sending special information.

[0079] The last of a packet header is rounded off about each packet header in the byte boundary. Therefore, in order to send information special whenever rounding-off is required, 1 to 7 bits is used. Similarly, each packet is rounded off in the byte boundary, and, thereby, prepares 1 to 7 bits (it is assumed that slight roundness is required). Moreover, several bytes can be extended by the packet of the last in a tile part. These special cutting tools may be used for sending an addition and information.

[0080] The die length of the compressed data to a code block can be given into

the packet header of the expression which is not min. Expressional selection (for example, the non-minimum expression) can be used in order to direct other information.

[0081] The packet header of JPEG2000 specification uses a tag tree about concealment of tag tree data for coding of the 1st inclusion and zero bit side information. When there are two or more code blocks, tag trees seem to be four trees of the minimum value. For example, in 16 code block in the 4x4 arrangement in a packet, arrangement is as follows.

[0082]

[Equation 3]

```

10  7 12 15
 3 20 21  5
81 45  5  9
18  8 12 24

```

The example of the tag tree which is the minimum one to 4x4 above-mentioned arrangement is as follows.

[0083]

[Equation 4]

```

3   0 2   7 4   7 10
   5 2   0 17 16 0
      73 37 0 4
      10 0 7 19

```

Here, "3" is added to the value of each code block, and, as for "0", "2", "5", and "2", each is added to four corresponding code blocks. Finally, there is a value for every code block. That is, the minimum tag tree is formed by taking 2 x2 groups of the beginning in above-mentioned 4x4 arrangement, and it is to find the minimum value out of four values. In this case, it is [0084] to 2 x2 Brock.

[Equation 5]

$$\begin{array}{cc} 10 & 7 \\ 3 & 20 \end{array}$$

It comes out, and it is and the minimum value is 3. This is performed also to other 2 x2 Brock. And these identified minimum values are again estimated by this example, in order to determine those minimum values that are "3." And it subtracts from the four minimum values and the minimum value is following [0085].

[Equation 6]

$$\begin{array}{cc} 0 & 2 \\ 5 & 2 \end{array}$$

It forms. And to the remaining numbers in 4x4, several 3 is subtracted from each value with the value in 2 x2 corresponding to the specific value in 4x4

arrangement, and, thereby, becomes an above-mentioned tag tree. The first line is added as follows.

$10=3+0+77=3+0+412=3+2+715=3+2+10$ variable length code may be efficiently used for expressing a small number.

[0086] The example of the tag tree which is not min is as follows.

[0087]

[Equation 7]

2	1	3	7	4	7	10
	6	3	0	17	16	0
			73	37	0	4
			10	0	7	19

(It is cautious of the ability of "2", "1", "3", "6", and bit stream data fewer than "3" to be used to express "3", "0", "2", "5", and "2".)

If a tag tree expression is made, based on whether zero are in 2 x2 Brock, the decision to which an expression is related for whether being min or not can once be made. Therefore, this information is hidden. For example, it can be used although one bit block shows one in 2 x2 Brock and it is shown that **** is not some minimum tag trees, however in order to send some specific information to a decoder. Similarly, if 2 is min within 2 x2 Brock, such a fact can send

information which is different in a decoder.

[0088] JPEG2000 -- POC, QCD, and a QCC marker can have a redundant entry

A code stream is quantized and the marker seems not to re-write it in. For example, QCD and a QCC marker have a value over some subbands specified according to the functor of a marker. When few subbands are actually encoded by the inside of a bit stream, data may be hidden in the value used in order to make a subband *****. Or it was permuted and hidden, it is used for side band information and deals in a redundant entry.

[0089] Or it was hidden, side band information may also include after-treatment directions (for example, or it Sharp-izes that tile with a specific filter, optical character recognition (OCR) is performed to emphasis, smoothing, or this field), decode directions, security (for example, cryptographic key for decoding a remaining image or other remaining images etc.), code stream discernment (for example, POTUS label attachment as cage JINETA of a file etc.), and/or other information.

[0090] The use hierarchy of the hierarchy at the time of coding is a part of JPEG specification. In the one example, probably, the side band information on a COM marker is used by the decoder, in order to enable a hierarchy's selection during

decode. This side band information is used for choosing the hierarchy for the quantization after compression in order to make the rate / distorted target over the contents of a frequency from which it differs for the field of a different viewing-and-listening distance, different resolution, and a different interest, and analysis (as [find / for example, / the edge of a text]) agree.

[0091] A hierarchy is beforehand defined by one example based on a rate. It seems that for example, the 1st hierarchy expresses 1-bit per pixel image, and the 2nd hierarchy, on the other hand, says that he expresses 2-bit per pixel image. Therefore, a hierarchy changes from the lowest quality to the highest quality. Similarly, a target rate can also be doubled with a low resolution.

[0092] Side band information is accumulated in the marker segment in a code stream. In the one example, a JPEG2000 comment (COM) marker is used so that the information about a hierarchy may be offered. Especially a COM marker is used for the byte count about each resolution and/or the rate covering a whole image or the relative byte count to the hierarchy of each addition being shown. Table 6 goes over the tile in an image, is a byte count and shows each class and its resolution. Such a table has a distortion value instead.

[0093]

[Table 6]

表 6

lev=0	layer=0	comp=0	bytes=529
lev=0	layer=0	comp=1	bytes=555
lev=0	layer=0	comp=2	bytes=493
lev=0	layer=1	comp=0	bytes=129
lev=0	layer=1	comp=1	bytes=130
lev=0	layer=1	comp=2	bytes=123
lev=0	layer=2	comp=0	bytes=7
lev=0	layer=2	comp=1	bytes=8
lev=0	layer=2	comp=2	bytes=12
lev=0	layer=3	comp=0	bytes=1
lev=0	layer=3	comp=1	bytes=1
lev=0	layer=3	comp=2	bytes=129
lev=1	layer=0	comp=0	bytes=705
lev=1	layer=0	comp=1	bytes=898
lev=1	layer=0	comp=2	bytes=712
lev=1	layer=1	comp=0	bytes=146
lev=1	layer=1	comp=1	bytes=114
lev=1	layer=1	comp=2	bytes=116
lev=1	layer=2	comp=0	bytes=224
lev=1	layer=2	comp=1	bytes=250
lev=1	layer=2	comp=2	bytes=263
lev=1	layer=3	comp=0	bytes=201
lev=1	layer=3	comp=1	bytes=212
lev=1	layer=3	comp=2	bytes=200
lev=2	layer=0	comp=0	bytes=889
lev=2	layer=0	comp=1	bytes=1332
lev=2	layer=0	comp=2	bytes=1048
lev=2	layer=1	comp=0	bytes=240
lev=2	layer=1	comp=1	bytes=329
lev=2	layer=1	comp=2	bytes=328
lev=2	layer=2	comp=0	bytes=599
lev=2	layer=2	comp=1	bytes=767
lev=2	layer=2	comp=2	bytes=725
lev=2	layer=3	comp=0	bytes=335
lev=2	layer=3	comp=1	bytes=396
lev=2	layer=3	comp=2	bytes=420
lev=3	layer=0	comp=0	bytes=1
lev=3	layer=0	comp=1	bytes=395
lev=3	layer=0	comp=2	bytes=402
lev=3	layer=1	comp=0	bytes=251
lev=3	layer=1	comp=1	bytes=450
lev=3	layer=1	comp=2	bytes=562
lev=3	layer=2	comp=0	bytes=525
lev=3	layer=2	comp=1	bytes=990
lev=3	layer=2	comp=2	bytes=1313
lev=3	layer=3	comp=0	bytes=1214
lev=3	layer=3	comp=1	bytes=1798
lev=3	layer=3	comp=2	bytes=2585

In other examples, alignment is every hierarchy. Thus, the following hierarchy =0, cutting tool =7959, bit rate =0.971558, alignment according [above-mentioned information] to PSNR=30.7785, hierarchy =1, cutting tool =10877, bit rate =1.327759, alignment by PSNR=32.0779, hierarchy =2, cutting tool =16560, a bit rate = as shown in 2.021484 and alignment by PSNR=35.7321, it is unified to each level (not separated by level or the component).

[0094] The distortion by the hierarchy is based on PSNR. For example, hierarchy =0, PSNR=30.7785 hierarchy =1, PSNR=32.0779 hierarchy = it is 2 and PSNR=35.7321.

[0095] In the alternative example, such information is hidden in the code stream as mentioned above. This information may be used for controlling rate distortion.

[0096] A hierarchy is beforehand defined in a specific viewing-and-listening distance by other examples. In such a case, data are divided into a hierarchy from the highest frequency and the lowest resolution to the lowest frequency and the highest resolution.

[0097] In the one example, a hierarchy shows the sum total of the bit for which the whole image to the hierarchy and the hierarchy in front of all is continued.

(For example, 16,011 bits of total numbers of bits to a hierarchy 0 and a hierarchy 1 currently listed by the degree, a hierarchy 1, are shown.) Table 7 shows this kind of absolute rate information.

[0098] Table 8 shows relative rate information. A hierarchy 0 shows 4096 bits and a hierarchy 1 shows 11,915 etc. bits etc.

[0099]

[Table 7]

表7

階層	レート(バイト)
0	4,096
1	16,011
2	40,000
3	100,000
4	250,000
5	500,000
6	1,000,000
7	2,500,000
8	5,500,000

[0100]

[Table 8]

表8

階層	レート(バイト)
0	4,096
1	11,915
2	23,989
3	60,000
4	150,000
5	250,000
6	500,000
7	1,500,000
8	3,000,000

For example, when it is allowed that only 750,000 bytes decodes an image, (since 1,000,000 bytes shown to the hierarchy 6 of 500,000 bytes of a hierarchy 0-5 are included), all in which decode and it deals are the one half of a hierarchy 5 and the important hierarchy 6. The packet from a hierarchy 6 is not contained in some the examples. In other examples, some a hierarchy's 6 packets are contained and other packets are transposed to a zero packet so that the sum total of hierarchy 6 data may become about 250 and 0000 bytes.

[0101] Drawing 22 shows the example of the hierarchization to 3 level, MSE, or the same 5.3 irreversible conversion. Reference of drawing 22 shows 45 hierarchies. The hierarchy of each addition improves MSE in the sequence of giving the good Leto distortion, to MSE.

[0102] Drawing 23 shows other examples in which conversion has 5 level and is

divided even into the data hierarchy 0-3. By a hierarchy 0 corresponding to the thumbnail version, a hierarchy 0-1 corresponds to monitor (or screen) resolution, and a hierarchy 0-2 corresponds to print resolution, and a hierarchy 0-3 responds to no losing.

[0103] A hierarchy may be beforehand defined on some of other distorted criteria (for example, MSE, MSE by which weighting was carried out, clearness of a text, etc.) by the alternative example.

[0104] Since an image is generated, in order to choose a hierarchy, the information about a hierarchy is used for a decoder from a code stream. the decoder which knows the desirable viewing-and-listening environment (refer to Table 9) from application or activation, and uses the information from a ***** code stream for a hierarchy -- however, a code stream can be quantized in order to display an image in viewing-and-listening distance. Drawing 9 shows such a decoder. If drawing 9 is referred to, since the quantized code stream 904 is generated through choosing a suitable hierarchy, including the quantization logic 902 which receives a code stream and inspects a COM marker, a decoder 901 will use the information about the viewing-and-listening distance memorized in memory 903. Since image data 906 is generated, after the quantized code

stream 904 chooses a hierarchy, it is decoded by the decode logic 905 (for example, JPEG2000 decoder). A simple decoder only disregards the data within a comment marker.

[0105] Drawing 10 shows the flow of the processing for using a hierarchy at the time of decode. This processing is performed by hardware (for example, the logic of dedication, a circuit, etc.), software (it performs by the general purpose computer or the exclusive machine), or processing logic including those both combination.

[0106] If drawing 10 is referred to, this processing will be started by the processing logic which receives the code stream of the compressed logical data (processing Brock 1001). This image data is systematized by two or more hierarchies, and that each contains the encoded data which apply a vision value to an image in an image (for example, it is [which it is visible to Sharp] more clear, and contrast is more good). The following processing logic chooses one or the hierarchy beyond it based on side band information for quantization (processing Brock 1002). After selection, processing logic reverse-compresses the hierarchy by whom a code stream is not quantized (processing Brock 1003).

[0107] When some of tiles, tiles and edit 1 ** of a packet, and a code stream are

formed, it is desirable to edit some images. That is, for example, in order to form a code stream, the group of a tile is decoded, after encoding. After decoding the group of a tile, edit is performed and **** of a tile is encoded following it with what was edited into the same size as the tile encoded before decoding. The example of typical edit includes Sharp-izing of a text, and removal of "bloodshot eyes." Without rewriting the whole code stream, it is in memory or a JPEG2000 code stream can be edited within a disk file system.

[0108] Drawing 11 shows the flow of one example of edit processing. This processing is performed by hardware (for example, the logic of dedication, a circuit, etc.), software (it performs by the general purpose computer or the exclusive machine), or processing logic including those both combination.

[0109] If drawing 11 is referred to, processing logic will determine the tile and tile part which include first the field which should be edited, resolution, a component, and/or a partition, or a packet, and will decode them (processing Brock 1101). This decision is made by answering the user who chooses a field and/or resolution of operation. This decision may use the edit information which carries out whether the part into which which part or tile is edited is included with higher resolution in order to determine. Once decode is completed, processing logic will

perform desired edit (processing Block 1102).

[0110] After performing desired edit, processing logic represses data to the encoded data (processing Block 1103), and forms a permutation tile, a tile part, or the packet to a code stream (processing Block 1104). In the one example, in case a permutation tile, a tile part, or a packet is formed, processing logic pads a cutting tool's data at the last of a code stream, in order that they may form the permutation tile, tile part, or packet of the same size as a non-edited version, when new data are smaller than the non-edited version of data.

[0111] In the alternative example, the marker or tag like a COM marker segment of suitable die length may be used for processing logic instead of padding. A COM marker can include information [can use it in order to fill a tooth space, or] to include an encoder. For example, the copyright license about the side band information that it explained here, or an image, a text, or information like other fill format information should be included.

[0112] In the one example, the component of all which **** is omitted until data suit processing logic in a code stream on the occasion of formation of a permutation tile, a tile part, or a packet, when new data are larger than the non-edited version of data.

[0113] Edit of an image is performed by changing a tile, a tile part, or the encoded data to a code block. In the one example, edit is performed by quantizing instead of extending, without changing a file. In other examples, in order to enable the escape of a predetermined amount, the further space of a predetermined amount is assigned to per a tile or code block. Furthermore, in other examples, by operating a tile header, the encoded data are put on the last of a file, and arrange invalid tile data in a COM marker.

[0114] When there is a tile part of consecutiveness depending on the data in the part of the code stream currently edited, it is cautious of it becoming impossible to use these tile parts within a code stream. A decoder is told about directions of this data that cannot be used by one of some approaches. These approaches include inserting or changing information, in order to show the existence of data and/or the location which cannot be used. In the one example, in order to show that the data in the tile part which follows the edited tile part may be unable to use it, a condition buffer is used for application. A condition buffer may be in work-piece memory, and describes the dependency between packets. When a former packet is changed, a consecutive packet cannot be decoded as it is. or [that the packet of these consecutiveness is edited according to it] -- or it must

be deleted. In other examples, such directions may be made by generating the PPT marker in which it is shown that a part for the data division of those tile parts is made into zero, or there are no data.

[0115] In order to generate the quantized multiplier value, the division of the multiplier which are some or by which quantization from all subbands is not carried out during the optimal coding personal appearance child-ized coding is done by Q value. This Q value has a wide range value. A typical encoder quantizes some values within the value of the single range made equal to one single multiplier value. Essentially, all specific multipliers within the limits are quantized by the same value. This is illustrated by drawing 12 , the range of a value carries out owner *Perilla frutescens* (L.) Britton var. *crispa* (Thunb.) Decne. of the curve of a campanulate configuration, and all specific values within the limits like the range R1 are sent to a decoder as one quantized value like R1', and a decoder reconfigurates these values to a specific value. It is assumed that a decoder reconfigurates these values to a predetermined (for example like [here / supposing Q is a quantization step size / of floor ($1/2$ and $\min+1/2$, max) or $\min+1/2$, and Q]) value. For example, when the range of a value is between 16 and 31, a value can consider that a decoder is 24. In the one example, other

values like floor ($\frac{3}{8}$ and $\min + \frac{5}{8}$, max) or $\min + \frac{3}{8}$, and Q are chosen instead of using one half as a value. Here, Q is a quantization step size. Therefore, when the range is between 16 and 31, a decoder reconfigures the value to 22 instead of 24.

[0116] In some cases, two multipliers which adjoin spatially are numerically near mutually, but in drawing 12 , it may be in another quantization bottle (quantization range) like the multiplier value 1201 of the range R2, and the multiplier value 1202 of the range R1. Distortion may take place in the result of this quantization. In the one example, to the multiplier near the boundary of two quantization bottles (quantization range), an encoder chooses a bottle (range) like the range R1 where a multiplier like a multiplier 1201 is quantized, and, thereby, the coordination between adjoining multipliers like a multiplier 1202 is maintained. This is useful to avoiding distortion. That is, when it moves to a high bottle (range) from a bottle (range) especially with a small multiplier, this technique decreases distortion, and makes a rate increase, and it deals in it.

[0117] In applying wavelet compression to the reduction animation sequence of the flicker to Motion-JPEG, a flicker sometimes occurs. Or the inside of the field in a continuous frame becomes bright, the example of such a flicker may include

the appearance (mosquito noise near an edge) of the image which becomes dark, or edge change, as an animation sequence is reproduced. To the frame which an animation sequence follows, or, therefore, the application of different local quantization to the noise which gets worse by quantization to which it is viewed and listened temporarily may require the flicker.

[0118] In order to decrease a flicker, and it is in the same location within a continuous frame, the multiplier near the same value is made by the same value.

That is, the multiplier value in a continuous frame is set as a predetermined value. This is essentially the format of the quantization applied during coding.

Drawing 13 is the flow Fig. of one example of the processing which decreases a flicker.

[0119] The trial of whether to apply such quantization to the multiplier value in the frame which follows is due to the quantization performed for the multiplier in a front frame. Therefore, while a decoder decodes data independently for every frame, an encoder uses the dependency of a frame, in order to remove a flicker.

[0120] Based on mutual threshold relation [value / multiplier] in order to decrease the flicker in Motion-JPEG in the one example, it is changed (quantizing) and is **. For example, it is the following, when D_n and D_{n+1} are the

multipliers (the same subband as the same spatial position) to which it corresponds in two frames before quantization, D'_n and D'_{n+1} express these multipliers after quantization and $Q(-)$ is scalar quantity quantization, and when a

```

    if ( | Q (Dn+1) - (D' n) | < T)
        D' n+1 = D' n
    else
        D' n+1 = Q (Dn+1)

```

value T is a threshold,

It *****. For example, ***** [a value T / twice the quantization step size]. For a limit, other values of T are [0121] although not carried out.

[External Character 1]

$\sqrt{2}Q, 1.5Q, 2\sqrt{2}Q$

It comes out.

[0122] One of the multiplier values may be changed so that it may become one predetermined approximation of other multiplier values. This approximation is determined by a certain threshold. or [that a user sets up a threshold based on a certain criteria] -- or it is accommodative. Probably this threshold may differ from a subband based on the coordination (the number of frames with this near multiplier) of a specific value. In the one example, a multiplier value is set up

equally to other multiplier values. an alternative example -- a multiplier -- the inside of the quantization bottle (quantization range) size of other multiplier values -- or it is set up the twice of quantization bottle (quantization range) size.

[0123] Drawing 14 shows one example of the encoder (or the part) which performs above-mentioned quantization. If drawing 14 is referred to, a quantizer 1400 will receive the multiplier 1410 to the frame of the animation sequence from wavelet transform (not shown). The multiplier is received by the quantization logic 1401. The threshold memorized in memory 1403 to the multiplier value about the frame before memorizing in memory 1403 to the multiplier 1410 is compared with the scalar quantity child-ized value Q given from memory 1404.

[0124] The quantization logic 1401 may contain comparison (for example, logic [which has the gate], circuit, etc. are needed) hardware, or the software which performs a comparison. This comparison hardware and software perform a subtractor or subtraction actuation. This result is quantized by the code stream (suppose that some values were changed).

[0125] This is applied over two or the frame beyond it. Moreover, the comparison is not restricted to two continuous frames. A comparison may go over the frame of 3, 4, and 5 grades, in order to determine whether change exists or not. The

example in which drawing 24 is used for the value in the 1st and 3rd frame setting up the value in the 2nd frame is shown.

[0126] Quantization is noticed about the code stream quantization using the regulation based on a code block being sufficient.

[0127] In the rate control, quantization, and hierarchization 1 example, alternative quantization of a multiplier is performed during coding by setting up the subset of the amelioration bit which should be a symbol (MPS) with still higher possibility. This may be performed in respect of the bit by which the own alternative was made. For example, the amelioration bit set as MPS for the purpose of having a sharp text image while minimizing the encoded data which are demanded from a background image, when a text was in a background image does not affect the text to the last bit side, using the actual value over the bit which affects a text.

[0128] Such a quantization device is used for performing uneven quantization size. For example, when it desires to have a background in few bits, a setup of the amelioration bit to MPS can operate as a format of quantization. Although this quantization device generates distortion of a certain level, it lowers a bit rate required to transmit a code stream.

[0129] Although this technique is given to the bit generated between amelioration pass and being dealt in it, this technique is noticed about having the application to other compressor styles (for example, the list generated between low-ranking pass, the tail bit of CREW of Ricoh Silicon Valley of the MENRO park (Menlo Park) of California, MPEG IV texture mode, etc.).

[0130] In the one example, the same technique may be applied to other inter-frame change. That is, in the one example, change by the rate distortion in one frame is performed within a consecutive frame, in order to prevent distorted effectiveness.

[0131] The quantization which a user specifies is offered in the rate control and quantization 1 example. 3 level conversions of one component -- receiving -- level 1 HH, level 1 HL and LH, level 2HH, level 2HL ** LH, level 3HH, and level 3 -- it comes out enough with the quantization value of 7 of HL, LH, and level 3LH.

[0132] When a quantization value is the bit side to omit (equal to the formation of a scalar quantity child by the exponentiation of 2), in a great portion of application, 3-bit value (0 ... 7) is enough. (To an image component with 12 bits or the bit depth beyond it, 5, or the conversion level beyond it, probably more high quantization is useful.) A value 0...6 is used for specifying the number of the

bit sides to omit, and is used for meaning that 7 throws away all bit sides. A triplet value is written in the controller which controls compression (or reverse compression) hardware (for example, JPEG2000 compatible hardware), in order to perform quantization.

[0133] To 3 component color quantization, the separate value over each component can use 21 value. - brightness -- receiving -- 7 and the color difference -- receiving -- 14 values of 7 -- it can be used - brightness -- receiving -- 7 -- and 17 values of 5 can be used to the data by which the 4:1:1 subsample was carried out to each color difference component. - The duodecimal of 5 can be used to the data by which the 4:1:1 subsample was carried out to 7 and a color difference component to brightness. - 19 values of 6 can be used to the data by which the 4:2:2 subsample was carried out to 7 and each color difference component to brightness, and 13 values of 6 can be used to the data by which the 4:2:2 subsample was carried out to 7 and a color difference component to - brightness.

[0134] Since $21 \times 3 = 63$ bit is smaller than 8 bytes, transmitting or accumulating quantization uses few resources. A central-process unit (CPU) chooses one predetermined quantizer from a table, and writes it in CPU or other controllers

which control JPEG2000 hardware (chip) of a special purpose about each frame of a Motion-JPEG 2000 video sequence. Instead, one activation of JPEG2000 may have the small memory holding the quantizer with which 8 in which it is chosen to each frame and deals differs from 16.

[0135] A quantizer can be used also in order to assign a hierarchy a bit side. For example, the quantizer which specifies the bit side of the coding pass to quantize is sufficient as Q0, Q1, and Q2. Although a quantizer Q0 generates most many loss, on the other hand, a quantizer Q2 generates the fewest loss. A hierarchy 1 is data which were not quantized by Q1 but were altogether quantized by Q0. A hierarchy 2 is data which were not quantized by Q2 but were altogether quantized by Q1. A hierarchy 3 is data altogether quantized by Q2.

[0136] or [that simple quantization drawing 17 and 18 omit some multipliers LSB] -- or the quantizer (label A...Q) of instantiation to 3-level 5 / 3 conversion which is not encoded is shown. The omission of N bit side is equivalent to the 2-N formation of a scalar quantity child. The subband from which quantization changes is highlighted about a front quantizer at the rectangular head of a broken line. All the quantizers D, K, and Q have the same relation between subbands. Other better quantizers may be used about other distorted criteria,

concerning MSE.

[0137] Verilog (very log) which is the following of instantiation changes single quantization value "q" into seven quantizers (the number of LSB to omit). It is that variable q_1_HH is used to a level 1 HH multiplier, and variable q_1_H is used to a level 1 HL multiplier and LH multiplier etc. The value which some q follows is [0138].

[External Character 2]

$i \geq 0$

It comes out and is set to quantizer:0, 1 and 2 and 3, 4 [same] and 5 [same], $8i+6$, and $8i+7$ to some integers i of all.

[0139]

[Equation 8]

```

module makeQ(q, q_1HH, q_1H, q_2HH, q_2H, q_3HH, q_3H,
q_3LL);
    input [5:0] q;
    output [3:0] q_1HH;
    output [3:0] q_1H;
    output [3:0] q_2HH;
    output [2:0] q_2H;
    output [2:0] q_3HH;
    output [2:0] q_3H;
    output [2:0] q_3LL;

    wire [3:0] temp_2H;
    wire [3:0] temp_3HH;
    wire [3:0] temp_3H;
    wire [3:0] temp_3LL;
    wire [2:0] qlo;
    wire [2:0] qhi;

    assign qlo = q[2:0];
    assign qhi = q[5:3];

    assign q_1HH    = qhi + ((qlo >= 2) ? 1 : 0);
    assign q_1H     = qhi + ((qlo >= 4) ? 1 : 0);
    assign q_2HH    = qhi + ((qlo >= 6) ? 1 : 0);
    assign temp_2H  = qhi + ((qlo >= 1) ? 0 : -1);
    assign temp_3HH = qhi + ((qlo >= 3) ? 0 : -1);
    assign temp_3H  = qhi + ((qlo >= 5) ? 0 : -1);
    assign temp_3LL = qhi - 1;

    assign q_2H    = (temp_2H < 0) ? 0 : temp_2H;
    assign q_3HH   = (temp_3HH < 0) ? 0 : temp_3HH;
    assign q_3H    = (temp_3H < 0) ? 0 : temp_3H;
    assign q_3LL   = (temp_3LL < 0) ? 0 : temp_3LL;

endmodule

```

The weighting table 9 of human being's vision system to a color and a frequency shows the bit side of the addition to quantize (for example, cut-off) to the brightness using the frequency response of human being's (from the table J-2 JPEG2000 specification's) vision system. Probably, the viewing-and-listening distance of 1000 pixels will be suitable to view and listen to the image on a computer screen. Probably, a longer viewing-and-listening distance will be

suitable to a printing image or television.

[0140]

[Table 9]

表9-輝度に対する人間の視覚システムの重み付け

サブバンド	視距距離に対する量子化する余分なビット面 ...		
	1000 画素	2000 画素	4000 画素
1HH	2	4 or 5	全て捨てる
1HL, 1LH	1	2 or 3	6
2HH	—	2	4 or 5
2HL, 2LH	—	1	2 or 3
3HH	—	—	2
3HL, 3LH	—	—	1

Furthermore, the color difference may be quantized still more heavily than brightness.

[0141] Drawing 19 is started by drawing 17 (D), and (as opposed to both brightness and the color difference) frequency weight is applied to a 1000-pixel child long distance, it holds so that 3LL color difference may not be changed, and 1HL and 1HH color difference about 4:2:2 are thrown away, and the quantization into which an additional 2-bit side is thrown away to the remaining color difference is shown.

[0142] The sharp text without a ringing strain is more desirable than the exact gradation value over a text/background. That is, gradation level is 50% (for example), and supposing it is inserted at 60%, when an image is a text, it may

often be visually desirable. In the one example, LL (DC) multiplier is a low bit rate, and is quantized still more heavily to a text rather than a non-text image. For example, to a 8-bit image component, 8, 16, or the quantization step size of 32 may be used to the field of only a text, and it may be used to the field where 1, 2, or the quantization step size of 4 does not contain a text. This makes possible still higher fidelity to a high frequency multiplier, and serves as a text which has a sharp edge by that cause.

[0143] The use table 10 of the quantizer for dividing into a hierarchy shows the quantizer of instantiation of 16. The quantizer 15 was not lost. The quantizer 8 is the same as what is shown in drawing 19 . These can be used for dividing a subband bit side into a hierarchy.

[0144]

[Table 10]

表10

サブバンド	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Y1HH	全て	全て	6	6	5	5	4	4	3	3	2	2	1	1	0	0
Y1HL,LH	6	5	5	4	4	3	3	2	2	1	1	0	0	0	0	0
Y2HH	5	4	4	3	3	2	2	1	1	0	0	0	0	0	0	0
Y2HL,LH	4	4	3	3	2	2	1	1	0	0	0	0	0	0	0	0
Y3HH	4	4	3	3	2	2	1	1	0	0	0	0	0	0	0	0
Y3HL,LH	4	3	3	2	2	1	1	0	0	0	0	0	0	0	0	0
Y3LL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C ₁ HL,HH	4:1:1又は4:2:2に対してのみHLとHHは常に捨てられる															
C ₁ 1LH	全て	全て	全て	全て	6	6	5	5	4	4	3	3	2	2	1	0
C ₁ 2HH	全て	6	6	5	5	4	4	3	3	2	2	1	1	0	0	0
C ₁ 2HL,LH	全て	6	5	5	4	4	3	3	2	2	1	1	0	0	0	0
C ₁ 3HH	全て	6	5	5	4	4	3	3	2	2	1	1	0	0	0	0
C ₁ 3HL,LH	全て	5	5	4	4	3	3	2	2	1	1	0	0	0	0	0
C ₁ 3LL	全て	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C ₂ HL,HH	4:1:1又は4:2:2に対してのみHLとHHは常に捨てられる															
C ₂ 1LH	全て	全て	全て	全て	6	6	5	5	4	4	3	3	2	2	1	0
C ₂ 2HH	全て	6	6	5	5	4	4	3	3	2	2	1	1	0	0	0
C ₂ 2HL,LH	全て	6	5	5	4	4	3	3	2	2	1	1	0	0	0	0
C ₂ 3HH	全て	6	5	5	4	4	3	3	2	2	1	1	0	0	0	0
C ₂ 3HL,LH	全て	5	5	4	4	3	3	2	2	1	1	0	0	0	0	0
C ₂ 3LL	全て	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A hierarchy 0 contains all the data that separate with a quantizer 0 and are not quantized. This is only the 4-bit side of all the 3LL, 2HL, 2LH, 3HL, and 3LH, and brightness data excluding [removing the 5 bit side of 2HH] six bit sides of 1HL and 1LH except for 3HH(s). a hierarchy 1 is not in a hierarchy 0 and separated with the quantizer 1 -- it does not quantize -- it is and all data are included. The bit side 3 to bit side [to the receiving-1HL / except a 6 bit / as opposed to; and color difference 2HL, and 2LH and 3HH(s) except for a 5 bit / as opposed to color difference 3HL and 3LH in this / side / side /, and 1LH brightness bit side 5, and 2HH] 4, 3HL, and 3LH; it is all 3LL color difference. Finally, a hierarchy 15

contains LSB of 1LH color difference.

[0145] There are some known techniques about rate control of the rate control compression system accompanied by two or more hierarchies and tile parts. "into which the simple approach was compressed -- it is choosing distortion as every unit" (a unit is the subband of the frame within 8x8 Block of JPEG, and an animation sequence, the tile of a single image, and the tile in the image by which wavelet coding was carried out etc.). When a selected distortion draws a bit rate higher than a desirable average bit rate, it is increased to the new unit compressed by distortion allowed. When a selected distortion draws a bit rate lower than a desirable average bit rate, distortion allowed decreases to the new unit compressed.

[0146] Furthermore, a complicated approach carries out the buffer of the data compressed from some "units." The bit rate and/or distortion to each unit in each distorted level are memorized. And distortion which goes over all the units in a buffer and is allowed is determined when filling up with all buffers. A buffer can appear in the whole image being included enough, and, in a certain case, very high quality can be acquired. In JPEG2000, a hierarchy is designed so that the augment to quality may be included. Therefore, choosing distortion means

choosing the number of hierarchies used to each code block or a tile. Perfect explanation of rate control of this kind is David. It is indicated in IEEE paper "the high performance scalable picture compression accompanied by EBCOT" about the image processing of Taubman.

[0147] This processing has some faults. One fault is that the buffer memory to the whole code stream is the need. The 2nd fault is that the time amount until a code stream is outputted at the time of waiting) is high. The 3rd fault is that the 2nd pass uses a lot of time amount.

[0148] In order to ease these problems, each tile of a JPEG2000 code stream is encoded as mentioned above on at least 2 hierarchies. By completion of coding of each tile, the number of packets (for example, a hierarchy, resolution, a partition, a tile component) is outputted to a code stream as a perfect tile part. The remaining hierarchies are memorized in a buffer. The 2nd pass through the data with which the remainder in a buffer was encoded is an option. Between this 2nd pass, if space or degree between allows the excessive packet from each tile, it will be added to a code stream as a perfect tile part. In application of a fixed rate, only the packet in a predetermined rate is added. An only number of cycles are allowed in application of fixed time. One example of this processing is

shown in drawing 15 (A). Therefore, these are two perfect tile partial outputs to each tile.

[0149] Drawing 15 (B) shows some hierarchies and hierarchy 1-n. A hierarchy 1 is the output of the 1st pass, and possibility below fixed time or a fixed rate time limit is [the remaining hierarchies] high. The output in the 2nd pass in fixed time or a fixed rate demand is satisfactory for a hierarchy 2, attaining the same distortion over all components.

[0150] Since there is little data to process, the 2nd pass which above-mentioned processing made it possible to memorize some data with which the buffer was encoded, and the 1st data was outputted immediately (transmitting or memorizing), and let data pass has the dominance point of being high-speed. Moreover, a buffer ends by little memory.

[0151] The criteria as which which packet chooses [1st / of tile part] whether it constructs, and passes and goes are the same as that of other rate control algorithms. In the one example, the rate of a packet is smaller than the desirable average rate of the whole image. For example, when the final compression bit rate in 2.0bpp is desirable, the 1st pass arranges 1.5bpp(s) to each tile in a code stream, and arranges 1bpp for a buffer to each tile.

[0152] The 2nd pass chooses from the remaining data the packet arranged in the 2nd [of each tile] tile part. Therefore, in order to obtain 2.0bpp average coding, some tiles which have the high distortion after the 1st pass receive all the remaining data stored to the tile, and, on the other hand, other tile parts which have the low distortion after the 1st pass do not have data of the addition transmitted.

[0153] the rate control to the compressed code stream data -- some rate control techniques explained here include the rate control performed to the compressed code stream based on the demand performed by choosing some numbers of hierarchies held within a code stream. A parser may be used for generating the new code stream which shows a bit rate based on a hierarchy. This bit rate is equal to the bit rate specified by demand, or smaller than it.

[0154] A parser can use the DS called "packet structure" here. This DS is used for other purposes like the multi-purpose packet DS shown below, and it acquires and is careful of it. In the one example, this packet structure contains a packet initiation pointer and a packet size. Moreover, the partition where a tile number, resolution, a component, a hierarchy, and a packet belong is also included. Finally, it also has a selection flag. This flag shows whether it was

chosen within the array written out to a new code stream, when set as a predetermined value (for example, 1).

[0155] In the one example, a packet is read in the sequence which continues from a code stream based on the advance sequence information shown with the COD marker.

[0156] This byte count is calculated based on the bit rate desired by demand.

The number of the cutting tools belonging to a hierarchy 0 is added to the sum total. And the byte count of this sum total is compared with the number of desirable cutting tools. When the sum total is smaller than the number of desirable cutting tools, the hierarchy of one addition is added to the whole. By this processing being equal to the number of cutting tools with the desirable sum total, or becoming larger than it, all packets are continued until it is added.

[0157] Within structure, by the selection flag, those packets added to the sum total between this processing are marked noting that it is chosen.

[0158] Additional processing is suspended when the sum total is equal to the number of desirable cutting tools. When the sum total exceeds the number of desirable cutting tools, the packet in the hierarchy of the added last is deleted from the sum total. This is performed in order to guarantee that a bit rate is lower

than a desirable bit rate. Therefore, the packet lengthened from the sum total is marked as un-choosing into a subtraction step.

[0159] In the one example, a related marker like SOT, COD, and PLT is updated according to a demand. A packet is written in a new code stream. Packet structure may be formed using the following.

[0160]

[Equation 9]


```

typedef struct _PACK_ {          /* packet structure */
    int      start;    /* packet starting point */
    int      length;   /* packet length */
    unsigned short  t;   /* tile number the packet belongs to */
    unsigned short  r;   /* resolution the packet belongs to */
    unsigned short  c;   /* component the packet belongs to */
    unsigned short  l;   /* layer the packet belongs to */
    unsigned short  p;   /* precinct the packet belongs to */
    unsigned char   select; /* selection flag */
} Pack_t;

```

```

/* Store packets from tp->tile[i].Size[j] array to the packet structure array
**/* Layer progression (LRCP) order */

```

```

if(progression_order == 0){

    j = 0;
    for(i=0;i<number_of_tile;i++){
        m = 0;
        for(l=0;l<layer;l++){
            for(r=0;r<resolution+1;r++){
                for(c=0;c<component;c++){
                    for(p=0;p<precinct[r],p++){

                        tp->pk[j].start = tp->tile[i].pointer[m];
                        tp->pk[j].length = tp->tile[i].Size[m];
                        total_length += tp->tile[i].Size[m];

                        tp->pk[j].t = i;
                        tp->pk[j].r = r;
                        tp->pk[j].l = l;
                        tp->pk[j].c = c;
                        tp->pk[j].p = p;
                        m++;
                    }
                    j++;
                }
            }
        }
        num_packet[i] = m;
    }
}

```

If a multiple-purpose packet DS 1 ** packet is read in structure, it is possible to use the same DS as having mentioned above for making other analysis options easy.

[0161] In resolution analysis, the packet which should be excepted is marked as un-choosing. For example, 4 resolution code stream is given, and if it is that a demand generates 3-resolution code stream, a parser will mark all the packets belonging to resolution 4 as un-choosing. And the newly generated code stream contains only the packet from resolution 1 to resolution 3.

[0162] It is possible similarly to perform quality analysis to component analysis and advance conversion analysis for every step which processes a packet within the structure.

[0163] Packet DS can treat a complicated demand. For example, a demand requires a parser to generate 3-resolution, 2-hierarchies, and the code stream that has 1-component code stream.

[0164] The pixel often encoded finally is in the outside of the range of the basis of the allowed pixel from the regular bit depth as a result of the quantization performed to the clipping wavelet multiplier after each inverse transformation. Typically, these pixels are clipped in the range of original, and, thereby, the

original bit depth can be used for the further image processing or the further indicating equipment.

[0165] For example, a 8-bit image has the pixel values between 0 and 255 including it. After lossy compression is used, the decoded image may include a value like 256 as -5. In order to supply a 8-bit output, these values are clipped in 0 and 255, respectively. Since the original image did not contain the pixel of the outside of a clipping boundary, this clip procedure usually decreases the distortion about a pixel. This procedure is known and is recommended by JPEG2000 specification.

[0166] In addition to the boundary about a final output sample, there is a boundary of a value which can assume a multiplier in the various phases of wavelet transform. As it is in the outside of the original baseball world, quantization can change the changed wavelet multiplier partially, so that quantization can change the sample finally decoded, in order to make it be in the outside of the original boundary. When those multipliers clip on the original boundary of them, distortion decreases.

[0167] For example, the maximum of a low-pass multiplier is +191 after the horizontal (one dimension) 5-3 invertible transformation using 8-bit input sample

** specified by JPEG2000, and a min possible value is -191. A highpass multiplier must be the meantime which contains 255 as -255. After 1-dimensional perpendicular conversion, low - low (Low-Low) multiplier is -286 to 287. Therefore, when decoding a 8-bit image, a 1st level low-low-pass multiplier is generated (reverse wavelet transform from high level), a multiplier is clipped in -286 and +287, and distortion decreases. Similarly, after the 1st level perpendicular inverse transformation is performed, a low-pass multiplier is clipped in -191 and +191, and a highpass multiplier is clipped in -255,255.

[0168] The maximum and the minimum value to a multiplier differ from each other to each subband, each filter, each conversion level, and each image depth. These maximums and minimum values are calculable by finding the signal which draws max and min, performing a positive compression system, and recording maximum. This signal that draws extremal value comes from the input each pixel of whose is either the greatest or the minimum. which pixel which pixel should be max and should be min collapses the sequence which is -1, when a wavelet multiplier is negative -- and when a wavelet multiplier is negative, it can determine by collapsing the sequence which is +1. To 5-3 filter used by JPEG2000 PERT I, the target low-pass signal is $[-1+1+1+1-1]$, and a highpass

signal is [-1+1-1].

[0169] The signal (image) which generates largest LL value is [0170].

[Equation 10]

+1 - 1 - 1 - 1 +1
- 1 +1 +1 +1 - 1
- 1 +1 +1 +1 - 1
- 1 +1 +1 +1 - 1
+1 - 1 - 1 - 1 +1

Come out, it is, and +1 must be permuted by input maximum (for example, 255)

here, and -1 must be permuted by input maximum (for example, 0).

[0171] In order to determine maximum to an irreversible filter, it is enough for there to be no need of actually performing a system and to only collapse a wavelet multiplier. As opposed to five to reversible 3 filter, a floor function is used by count of a multiplier and it is used also for determining right maximum.

[0172] This is noticed about that it may be used to other filters (for example, 9-7 filter).

[0173] Drawing 28 is the flow Fig. of one example of processing for applying the inverse transformation accompanied by clipping to the multiplier changed partially. This processing is performed by hardware (for example, a circuit, logic of dedication, etc.), software (it performs by the general purpose computer or the

exclusive machine), or processing logic including those both combination.

[0174] If it returns to drawing 28 , processing logic will perform inverse transformation of the 1st level for a multiplier (processing Brock 2801). After that, processing logic clips the multiplier changed partially in the predetermined range (processing Brock 2802). Next, processing logic performs inverse transformation of the 1st level for the clipped multiplier (processing Brock 2803), and clips the multiplier changed partially in different predetermined range from the range in processing Brock 2802 (processing Brock 2804). Again, processing logic performs inverse transformation of the 1st level for the clipped multiplier (processing Brock 2805), and clips further the multiplier changed partially in other predetermined range (processing Brock 2806).

[0175] The typical decode processing containing the handling color management of the simplified color space is shown in drawing 25 . Reference of drawing 25 supplies the file which has a file format (for example, file format described in JPEG2000 specification) containing the restricted ICC profile to decode processing. Although the reverse compressed block 2501 reverse-compresses a file by performing reverse wavelet transform by taking the code stream part of a file and performing relational-model-izing and entropy

decode, it does not perform color conversion actuation. When a code stream shows RCT or an ICT component, conversion should be used for decoding a code stream and these are performed by Block 2502. That is, reverse RCT/ICT Block 2502 gets a component and "RCTY/N" directions (if it is yes and is RCT and no ICT), and performs specified inverse transformation, and supplies a RGB (it is non-display) pixel (when prescribed by functor, a reverse level shift is also performed).

[0176] With the information about a display, since an output pixel is generated, the ICC color profile from a file format is used for the last.

[0177] Reverse ICC Block 2503 receives a RGB (it is non-display) pixel and an ICC profile, and performs reverse color conversion to a display RGB pixel.

[0178] Drawing 26 shows one example of the camera encoder which is not suitable. If drawing 26 is referred to, a camera will generate a YCrCb pixel. A converter 2602 changes a YCrCb pixel into a RGB pixel, and supplies them to JPEG2000 typical encoder. An encoder has the RCT/ICT converter 2603 and a compressor 2604 follows it. A compressor generates ICCA to a code stream.

[0179] Drawing 27 shows one example of a simple camera encoder. That is, a simple camera encoder contains only compressor Block 2702 instead of RCT /

ICT transducer 2603, and a compressor 2604 being included. If drawing 27 is referred to, a camera 2701 will generate a YCrCb pixel and will supply them to a compressor 2702. A compressor generates the ICCB code stream accompanied by RCT equal to 1 (accompanied by the functor which shows that reverse RCT should be used on the occasion of decode) including JPEG2000 encoder without RCT conversion. The relation between ICCB and ICCA is the following formulas and [0180].

[Equation 11]

$$ICC_B = ICC_A \circ YCrCb^{-1} \circ RCT$$

It is alike and is given more. It is here and is [0181].

[External Character 3]

o

** and the composition of functions are shown.

[0182] The restricted ICC profile is the "functor" over the function of a pixel. A camera writes in the same profile to all pixels typically, and ICCB is calculated off-line by that cause, and it is copied to each output profile. HW about YCrCb-1 and RCT/ICT which operate about each pixel with the conventional technique is

indispensable.

[0183] Typically, coding JPEG2000 specification is used for treating the data of 4:4:4 formats as 4:4:4 data accompanied by quantization for 4:2:2 and 4:1:1 data. It cannot describe how the data of 4:1:1 or 4:2:2 formats are reconfigured in 4:4:4 formats for an output. In the one example, when encoding 4:1:1 data, an encoder treats 1HL, 1LH, and 1HH multiplier as zero. When encoding 4:2:2 data, an encoder treats 1HL and 1HH multiplier as zero. therefore, all the data in an excessive subband are quantized to zero -- having -- a decoder -- the -- a code stream is receivable so that it may expect. In other words, the encoded data resemble 4:4:4 data quantized heavily.

[0184] Two or more images in a thumbnail, a monitor, a printer, total resolution, and the resolution of file sequence plurality to quality are important in the situation of many image processings. Depending on application, a user may desire a thing from which different resolution differs and for which image selection is made. For example, a thumbnail image is used as an index of many images, and it deals in it. Moreover, a screen resolution image may be used for sending to the monitor which displays it. The low resolution to printer application is sufficient as a print resolution image.

[0185] The version from which the code stream of an image is systematized by the section and an image like for example, the thumbnail version, the screen version, the print version, and a non-lost version differs by that cause in the one example is continuous by quality.

[0186] A packet consists of one example so that a certain packet may correspond to specific resolution like a thumbnail. The combination of other packets and these packets expresses a monitor resolution image, and when combined with other packets, it expresses the printer version etc. POC and a tile part are used and grouping of both the parts of a code stream is carried out. For example, grouping of all the tiles of both of thumbnail size is carried out, and, for the tile of other resolution, the tile of other resolution is following a degree etc. following a degree. Drawing 21 shows advance of instantiation which carries out a single transducer pair. Grouping of the thumbnail of each tile is carried out into a tile part at the head of a file. Drawing 21 A shows that only the tile part 2101 is a part used to a thumbnail image. Drawing 21 B shows that the tile part 2102-2104 is contained with the tile part 2101 to monitor resolution. Drawing 21 C shows that the tile parts 2105 and 2106 are contained with the tile part 2101-2104 to printer resolution. Finally, drawing 21 D shows that the three

remaining tile parts 2107-2108 are contained with other tile parts to the non-lost version of data. a tile part -- these -- constructing -- it is arranged on a server in order of this advance.

[0187] One example of the processing accessed to that to which grouping of the tile part was carried out is shown in drawing 16 . The processing is performed by hardware (for example, the logic of dedication, a circuit, etc.), software (it performs by the general-purpose computer system or the exclusive machine), or processing logic including the both combination. In the following steps, an image is changed on the hierarchy who divides an image into sufficient resolution level and four sizes.

[0188] If drawing 16 is referred to, processing logic will opt for right resolution and hierarchization to a thumbnail first (processing Brock 1601). In the one example, in order to opt for right resolution and hierarchization to a thumbnail, processing logic forms POC restrained by that resolution and hierarchy to each tile, forms **** of a tile part, and arranges this POC to each tile in a code stream.

[0189] Next, it repeats processing Brock 1601 to monitor resolution noting that processing logic already has a thumbnail packet in a code stream (processing Brock 1602). And it repeats processing Brock 1601 to printer resolution noting

that processing logic already has a monitor packet in a code stream (processing Brock 1603).

[0190] Finally, a POC marker is formed in the edge of resolution and a hierarchy to each tile (processing logic 1604). In the one example, when forming a POC marker in the edge of resolution and a hierarchy forms the remaining tile part for four **** of a tile part, and non-lost conversion, it performs.

[0191] If the specific sequence of the packet defined in POC is important, it is noticed about it being a mere limit that there is nothing.

[0192] Computer system drawing 20 of instantiation is the block diagram of the computer system of instantiation. If drawing 20 is referred to, a computer system 2000 may have the client 150 of instantiation, or server 100 computer system. A computer system 2000 has the processor 2012 connected to the bus 2011 in order to process the transmitter style or the bus 2011, and information for communicating information. A processor 2012 has microprocessors, such as Pentium™, PowerPCTM, and Alpha™, for example, although a limit is not carried out to a microprocessor.

[0193] A system 2000 has further the random access memory (RAM) or other dynamic memory equipments 2004 (it is called the main memory) which were

connected to the bus 2011 which memorizes the information which boils a processor 2012 and is performed more, and an instruction. Main memory 2004 may be used for memorizing a temporary variable or other middle information, while boiling a processor 2012 and executing an instruction more.

[0194] a computer system 2000 also has data storage 2007 like the read-only memory (ROM) and/or other static memory equipments 2006 which were connected to the bus 2011, a magnetic disk or an optical system, and its corresponding disk drive, in order to memorize the static information and the instruction which a processor 2012 is looked like [instruction] and received. Data storage 2007 is connected to the bus 2011 which memorizes information and an instruction.

[0195] Further, a computer system 2000 has a display 2021 like the cathode-ray tube (CRT) or liquid crystal display (LCD) connected to the bus 2011, in order to display information on a computer user. The alphabetic-character input unit 2022 containing an alphabetic character and other keys is also connected to the bus 2011 in order to communicate the information and the select command which pass through a processor 2012. The further user input equipment is cursor control 2023 like a mouse, a trackball, a trackpad, a stylus, or a cursor arrow key

which communicates a select command with direction information to a processor 2012, and controls migration of cursor on a display 2021.

[0196] Other equipments connected to the bus 2011 are hard copy units 2024, and this is used for printing an instruction, data, or other information to paper, a film, or the medium of the same format. Furthermore, the sound record and the regenerative apparatus like a sault peaker and/or a microphone may also be connected to the bus 2011 which carries out an audio interface with a computer system 2000 as an option. Other equipments connected are the cable / radio function 2025 which communicates with a telephone or pocket equipment bus 2011.

[0197] It is cautious of the hardware relevant to the component of all which **** being used for a system 2000 among this inventions. However, other configurations of a computer system are some, or it will be understood that all equipments may be included.

[0198] Although it will become clear to this contractor that this invention has many deformation and amelioration if the above-mentioned explanation is read, don't understand the explained specific example as what restricts this invention. Therefore, the claim of this invention which is enumerating only the

configurations considered that reference of detailed various examples is indispensable to this invention is not restricted.

[0199]

[Effect of the Invention] The improvement of the compression and the reverse compressor style which enable the high speed, the low cost, the small memory, and/or the abundant activation of the description by the selections in JPEG2000 being used for this invention can be made.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram of JPEG2000 specification decode device.

[Drawing 2] It is drawing showing one example of the configuration of the image in memory.

[Drawing 3] It is drawing showing the configuration of the conversion memory to the various level which shows notionally how a multiplier is memorized to positive conversion.

[Drawing 4] It is drawing showing the example of the single memory which can memorize the various decomposition level of input image data and image data during positive conversion and inverse transformation.

[Drawing 5] It is drawing showing one example of processing treating input data.

[Drawing 6] (A) is drawing showing the system which has an advance sequence conversion parser, and (B) is drawing showing the advance converter changed into a quality advance code stream from a resolution advance code stream.

[Drawing 7] (A) is drawing showing two or more paths of changing a code stream into other advance sequence from one advance sequence, and (B) is drawing showing one example of the simplified conversion path which changes a code stream into other advance sequence from one advance sequence.

[Drawing 8] It is drawing showing one example of the processing which performs advance sequence conversion.

[Drawing 9] It is drawing showing the decoder which chooses the part of a code stream based on side band information.

[Drawing 10] It is drawing showing the flow of the processing which uses a hierarchy at the time of decode.

[Drawing 11] It is drawing showing the one example flow of edit processing.

[Drawing 12] It is drawing showing the bell mold curve of the range of the value quantized by the specific value.

[Drawing 13] It is drawing showing one example of the processing which decreases a flicker.

[Drawing 14] The encoder which performs quantization which decreases a flicker (or it is drawing showing one example of the part.)

[Drawing 15] (A) is drawing showing the processing which performs rate control,

and (B) is drawing showing some hierarchies of instantiation who can receive the 1st and the 2nd pass.

[Drawing 16] It is drawing showing one example of the processing which accesses the group of a tile part.

[Drawing 17] It is drawing showing the quantizer to 3 level 5 and one component to 3 conversion.

[Drawing 18] It is drawing showing the quantizer to 3 level 5 and one component to 3 conversion.

[Drawing 19] It is drawing showing the example of the quantization by which HVS weighting was carried out.

[Drawing 20] It is drawing showing Brock of one example of a computer system.

[Drawing 21] It is drawing showing the advance of the instantiation accompanied by a part to a single server.

[Drawing 22] It is drawing showing the example of the hierarchization to 5 and 3 irreversible conversion.

[Drawing 23] Conversion has 5 level and data are drawing showing the example currently divided into the hierarchy 0-3.

[Drawing 24] It is drawing which is used for the 1st and the value of the 3rd frame

setting up the value of the 2nd frame and in which showing an example of the situation that a flicker may be avoided.

[Drawing 25] It is the block diagram of decode processing of the conventional technique including the handling of a color.

[Drawing 26] It is drawing showing one example of the camera encoder which is not desirable.

[Drawing 27] It is drawing showing one example of a simpler camera encoder.

[Drawing 28] It is drawing showing the flow of one example of the processing for performing inverse transformation which clips for the multiplier changed partially.

[Description of Notations]

101 Data Alignment Brock

102 Algebraic-Sign-ized Machine

103 Multiplier Bit Modeling Brock

104 Quantization Brock

105 ROI Brock

106 Conversion Brock

107 Component Conversion

201 Memory

202 Wavelet Transform Processing Logic

202A Memory access logic

202B Wavelet transform

210 Tile

301 Input Tile

310 311 Input row

501 Memory

5021-502N Multiplier buffer

503 Relational Model

504 Entropy Encoder

505 Coding Data Memory

601 Parser

602 Server

603 All Images

604 Memory

901 Decoder

902 Quantization Logic

903 Memory

904 Code Stream

905 Decode Logic

1400 Quantizer

1401 Quantization Logic

1403 Memory

1404 Memory

1410 Multiplier

2000 Computer System

2007 Data Storage

2011 Transmitter Style or Bus

2012 Processor

2021 Display

2022 Alphabetic-Character Input Unit

2023 Cursor Control

2024 Hard Copy Unit

2025 Cable / Radio Capacity

2501 Reverse Compressed Block

2502 Reverse RCT / ICT Brock

2503 Reverse ICC Brock

2602 Converter

2603 RCT/ICT Converter

2604 Compressor

2702 Compressor Brock